

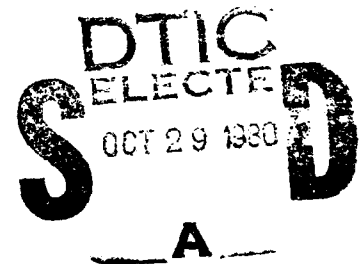
**FINAL REPORT: FY 79 SOFTWARE ACQUISITION  
PROCESS MODEL TASK**

**BY OSCAR SHAPIRO AND JOHN B. GLORE**

**JULY 1980**

**Prepared for**

**DEPUTY FOR TECHNICAL OPERATIONS  
ELECTRONIC SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
Hanscom Air Force Base, Massachusetts**



**DDC FILE COPY**

Approved for public release;  
distribution unlimited.

**Project No. 5720  
Prepared by  
THE MITRE CORPORATION  
Bedford, Massachusetts  
Contract No. F19628-80-C-0001**

**80 10 27 083**

**AD A091031**

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

#### REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

*Ralph P. Dunlevy*

RALPH P. DUNLEVY, 2Lt, USAF  
Project Manager

FOR THE COMMANDER

*Charles J. Grewe, Jr.*

CHARLES J. GREWE, Jr., Lt Col, USAF  
Director, Computer Systems Engineering  
Deputy for Technical Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
18 1. REPORT NUMBER ESD-TR-80-129-REV-1	2. GOVT ACCESSION NO. AD-A091031	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) <del>FINAL REPORT</del> FY 79 SOFTWARE ACQUISITION PROCESS MODEL TASK. Revision 1.		5. TYPE OF REPORT & PERIOD COVERED 9 FINAL REPORT 1979	
6. AUTHOR(S) 10 Oscar Shapiro John B. Glore		7. PERFORMING ORG. REPORT NUMBER 14 MTR-3871-Rev-1	
8. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P. O. Box 208 Bedford, MA 01730		9. CONTRACT OR GRANT NUMBER(S) 15 F19628-80-C-0001	
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Technical Operations Electronic Systems Division, AFSC Hanscom AFB, MA 01731		10. PROGRAM ELEMENT, PROJECT, TASK & A & WORK UNIT NUMBERS 16 Project No. 5720	
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 229		12. REPORT DATE 11 JULY 1980	
13. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		13. NUMBER OF PAGES 227	
14. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
15. SUPPLEMENTARY NOTES		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. KEY WORDS (Continue on reverse side if necessary and identify by block number) ACQUISITION MODELING SIMULATION SOFTWARE			
17. ABSTRACT (Continue on reverse side if necessary and identify by block number) This final report on the FY 79 Project 5220 Software Acquisition Process Model Task (522F) presents the approach taken to process model definition, quantification, and simulation; accomplishments; and status at the fiscal year's end. The report also identifies desirable improvements and outlines a plan for their incorporation and application in successive process model versions. ✓			

(over)

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

235050

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

↓  
**29. ABSTRACT (concluded)**

The report contains diagrams that represent the Full-Scale Development Phase of the Major System Acquisition Life Cycle. These diagrams can provide useful reference material for anyone who is (or expects to be) engaged in the planning or monitoring of a system development contract.

Revision 1 eliminates references to several documents not in the public domain. These changes, made to secure public release, are all minor. In addition, minor errors have been corrected.  
↑

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**



## ACKNOWLEDGEMENTS

The work described in this report was funded by the Computer Systems Engineering Directorate (TOI), Electronic Systems Division (ESD), Air Force Systems Command (AFSC). Major Neil D. McQuage, ESD/TOIT, was Project Officer. He and other members of the ESD Software Cost Estimation Working Group (ESD SCEWG), especially Captain Joseph A. Duquette, ESD/ACCE, and Mr. A. N. Sukert, RADC/ISIS, helped to define the scope of the work and provided several useful suggestions.

Credit for the original idea of a software acquisition process model is due principally to Mr. Gerard A. Bourdon. Mr. William J. White strongly influenced the early task definition. Ms. Donna M. Cosgrove helped to develop the process model concept further, assessed its feasibility, and did much of the early FY 79 work. Ms. Marie McLane is responsible for most of the process model simulation program prepared to date. Ms. Brooke Marshall and Ms. Carol Sanderson helped to calibrate process model estimates, to review process flow diagrams, and to proofread other document material. Ms. Lisa A. Popp typed and helped edit the many drafts.

✓

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525</

## TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
	LIST OF ILLUSTRATIONS	7
	LIST OF TABLES	8
1	INTRODUCTION	9
	1.1 POTENTIAL USES	10
	1.2 PHASED DEVELOPMENT APPROACH	11
	1.3 ORGANIZATION	12
2	SUMMARY	14
3	APPROACH	15
4	PROCESS MODEL DEFINITION	18
	4.1 BASIC PREMISES	18
	4.1.1 Conformance to Military Standards	18
	4.1.2 System, Segment, and CPCI Relationships	18
	4.1.3 Validation Phase Activities	19
	4.1.4 Support Facilities	19
	4.1.5 Phased Implementation Provisions	19
	4.1.6 Incidental Activities	20
	4.1.7 Resource Utilization	20
	4.2 PROCESS FLOW DIAGRAMS	22
	4.2.1 Global View	23
	4.2.2 Simulation Level Diagrams	23
	4.2.2.1 The High-level Simulation (HiSim) Diagram	24
	4.2.2.2 The Low-level Simulation (LoSim) Diagram	24

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
4.2.3 Expanded Views and Amplification Notes	25
4.3 PRIOR PROJECT EXPERIENCE BASE	25
5 PROCESS QUANTIFICATION	32
5.1 SOFTWARE DEFINITION FOR A TYPICAL SYSTEM	32
5.2 PARAMETER SELECTION	33
5.2.1 Activity Box Parameters	33
5.2.2 Decision Box Parameters	34
5.2.3 Special Event Box Parameters	34
5.3 PARAMETER VALUE ESTIMATION	34
5.4 PARAMETER VALUE CALIBRATION	35
5.4.1 Activity Box Timing Calibration	35
5.4.2 Manpower Assignment Calibration	36
5.5 FUTURE ACTIONS	36
5.5.1 Review by a Panel of Experts	37
5.5.2 Training Course Use and Follow-Up	37
5.5.3 Sensitivity Analysis	37
5.5.4 Analysis of Formally Collected Data	38
6 PROCESS SIMULATION	51
6.1 DEVELOPMENT FACILITY AND SIMULATION PROGRAMMING LANGUAGE SELECTION	51
6.1.1 Facility Selection	51
6.1.2 Simulation Programming Language Selection	52
6.2 DESIGN OVERVIEW	53

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
6.3 DATA BASE	54
6.3.1 List Structure	55
6.3.2 Data Packing	55
6.3.3 Data Base Size	56
6.4 DATA INPUT PROCESSOR	56
6.5 SIMULATION CONDUCT PROCESSOR	56
6.5.1 Event Notice Concept	57
6.5.2 Event Notice Use	57
6.5.3 Simulation Conduct Operation	57
6.5.4 Simulation Results Data Collection	58
6.5.5 Multi-pass Control	59
6.6 OUTPUT REPORT GENERATION	59
6.6.1 Model 0 Output Reports	60
6.6.2 Future Model Output Reports	60
7 ACCOMPLISHMENTS AND STATUS	65
7.1 ACCOMPLISHMENTS	65
7.2 CURRENT STATUS	65
7.2.1 Process Model Definition	66
7.2.2 Process Model Quantification	66
7.2.3 Process Model Simulation	66
8 ANTICIPATED GROWTH AND PLANS	67
8.1 GROWTH AREAS	67
8.1.1 Model Definition	67

## TABLE OF CONTENTS (Concluded)

<u>Section</u>	<u>Page</u>
8.1.2 Quantification	67
8.1.3 Simulation	68
8.1.4 Output Reports	70
8.2 PLANNED FY 80 ACCOMPLISHMENTS	70
8.3 OVERALL PLAN	71
8.3.1 Feasibility Analysis	72
8.3.2 Demonstration Model (M0)	72
8.3.3 Prototype Model (M1)	73
8.3.4 Initial Operational Model (M2)	74
8.3.5 Assessment of Initial Pilot Applications	74
8.3.6 Routine ESD Model Operation (M2 & M3)	75
8.3.7 Extended Model (M3)	75
8.3.8 Model Review and Evaluation	75
9 CONCLUSIONS AND RECOMMENDATIONS	78
APPENDIX A PROCESS FLOW DIAGRAMS AND AMPLIFICATION NOTES	79
APPENDIX B MODEL DEFINITION DATA	109
APPENDIX C RATIONALE FOR PROCESS MODEL DEVELOPMENT	137
APPENDIX D PARAMETER VALUE CALIBRATION	141
APPENDIX E SIMULATION PROGRAM LISTING EXTRACTS	151
APPENDIX F EVENT NOTICE SPECIFICATION	179
APPENDIX G DEMONSTRATION MODEL (M0) STATISTICS DEVELOPMENT	194
APPENDIX H DATA REPORTING FOR LATER SIMULATOR MODELS	201
GLOSSARY	223
REFERENCES	226

## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Major System Acquisition Process Normal Phases of Progression	27
2	Software Acquisition Process Model Activity Flow Overview - Full-Scale Development Phase	28
3	Software Acquisition Process Flow Diagram - HiSim Level	29
4	Model Box & Predecessor/Successor Counts	30
5	Phased Group Development Example	31
6	Typical Project - CPCI Breakdown	39
7	Typical Milestone Schedule for Operating Program	40
8	Typical Manpower Utilization Levels	41
9	Typical Manpower Costs for Operating Program CPCI	42
10	Activity Duration Calibration Summary - Main Path (Calibrated Activity Durations, with Iteration)	43
11	Program Acquisition: Activity Sequence and Timing Diagram	44
12	Data Base Structure	62
13	Data Base Size	63
14	Event Notice Processing Sequence	64
A-1	Flow Diagram Notation	82
A-2	Software Acquisition Process Flow Diagram - LoSim Level	84
A-3	Process Flow Expansion	96
D-1	Activity Duration Calibration	142
D-2	Activity Duration Calibration Summary	149

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Outline of Strawman Software Acquisition Process Model Development Plan	76
2	Tentative Later Process Model Version Characteristics	77
		83
A-1	Index to Figure A-2 Connectors & Box Numbers	102
A-2	Process Flow Diagram Amplification Notes	
B-1	Software Acquisition Process Model Network Linkage	116
B-2	Software Acquisition Process Model Activity Box Parameter Data	128
B-3	Software Acquisition Process Model Decision Box Parameter Data	134
B-4	Software Acquisition Process Model Counter & Special Event Box Parameter Data	136

## SECTION 1

### INTRODUCTION

This report documents the results achieved during FY 79 on Project 5220, task 522F, Software Acquisition Process Model. Since this is the final FY 79 report on this project, it includes previously reported findings to the extent necessary to make this report self-contained.

The concept of an ESD Software Acquisition Process Model grew out of earlier work in software cost estimation. Most of this prior work was based on an algorithmic or analytic approach by which computer program attributes (such as size, function, application, and complexity) along with developmental attributes (such as language, tools, methods, and experience) could be converted into cost estimates. As reported in a MITRE review of software cost estimation methods (CLAPJ76) the results of using these estimation methods on large military systems have been less than adequate.

More recently, a concept evolved which suggested that the software development process itself (rather than just overall attributes of the software and development methods) should be used as a basis for software estimating. While this approach might add some complexity to the estimating process, it appeared to have compensating advantages. As the idea was evaluated, it became clear that on military systems involving embedded software, the development process was too closely joined to acquisition procedures for the two to be separated. For this reason, we decided that the overall acquisition process (including both contractor and government activities) was a better basis for software estimation than development alone. A succinct statement of this idea and its rationale was provided in a July 1978 MITRE letter to ESD; the relevant portion of this is reproduced in Appendix C, Rationale for Process Model Development. This approach was later recommended within the Research Management Plan developed by the Air Force Systems Command Software Cost Estimating Working Group (SCEWG79). As stated in the letter, the MITRE evaluation

"strongly indicates substantial potential benefits from developing and applying in selected ESD-managed acquisition programs a simulation model of the software acquisition process. The type of software acquisition process model considered would represent explicitly (e.g., in flowchart form) the different Program Office (PO) and contractor activities that ESD-managed software acquisition entails, and how these different activities interact. It would represent activity sequences, repetitions of such sequences, alternatives, concurrency, and delays (e.g., due to waits for essential inputs). In this respect the model would somewhat resemble PERT, but without PERT's restrictions on loop representation, etc."

The concept involved early Model development and application to ESD-managed Electronic system programs, conducted per AFR 800-series regulations,



that include acquisition of embedded software. The clear need for improved ability to estimate better such systems' costs and schedules, and to manage better their development, guided the choice of this general application area. Later, the Model could be revised or extended for application to other AFSC Product Division programs, and to other types of systems. Another reason for the early emphasis was availability at MITRE of persons with extensive practical experience on such programs, able to develop the Model.

### 1.1 POTENTIAL USES

A number of advantages were seen for the Simulation Model. These include:

- a. Improved accuracy. This would result, in part, from the explicit contractual situation on which the Model would base its estimates.
- b. Measures of uncertainty, because the Simulation Model could produce a measure of estimates' dispersion, and corresponding estimate ranges, as well as point values.
- c. More credibility, because the estimates would be based on defined activities to which an acquisition program's management could relate and which they could understand.
- d. More flexibility, because the modeled process could include the effects of changes in development technique which will occur as the practice of software engineering matures.
- e. More versatility, because the Simulation Model could support many other uses (described below) than the generation of cost and time estimates. In fact, the concept and simulation program (i.e., the Simulator) could be applied to equipment procurement, total system acquisition, acquisitions conducted per different regulations, and many other processes.

While the main driving force behind the Simulation Model is the need for better software-related cost and schedule estimates, the Model could be effectively employed for other purposes, such as the following:

- a. The diagrams of the acquisition process would be useful for training military and support personnel for work on software acquisition. The simulation program (i.e., the Simulator) itself would also help the training, by presenting a dynamic picture which would illustrate the effects and consequences of alternative actions and decisions.
- b. The diagrams would be very helpful in project planning. They would provide a checklist that insured that important activities and products were not overlooked, and that contractual events and products were scheduled realistically; i.e., in conformance with the organic needs of the acquisition process. Past experience on many projects indicates that this need has often been overlooked, with dire consequences. The Simulator would also improve the

validity of the system planning trade-off analyses that are performed to establish the capability and capacity mix for a particular procurement.

c. The Simulator could be useful for evaluating contractor proposals to determine the extent to which the proposed schedule, allocated costs, development plans, etc. were consistent with Simulator forecasts, and thus with previous ESD experience.

d. During contract monitoring, the Simulator would help evaluate the consequences of milestone slippage, delays induced by Engineering Change Proposals (ECPs), ongoing cost reports vs. developmental progress, etc.

e. After the Model was put into routine use on ongoing contracts, and as the data associated with real contracts accumulated, the Model accuracy would improve as its processes and parameter values more closely reflected those found on real projects. As this happened the Model would evolve in concert with the improvements in the software development art. The resultant parameter changes would then provide an objective measure of the "trend line" and would enable more accurate future forecasts. At the same time, the data obtained on ongoing projects would enable the developmental performance on different contracts to be objectively and numerically compared.

f. The graphic description of the acquisition process provided by the Model would present a compact yet detailed view of how the Air Force obtains embedded software. This view could improve understanding of the process, and thus help to determine ways by which it could be improved. The objectives sought, for example, could be ways to reduce the overall time or costs, to obtain a more reliable product, or to establish the cumulative impact of the various system constituents (including operating functions, support functions, and data items) for use in tradeoff studies that would also consider each constituent's utility value. Use of the Simulator would allow the dynamics of the process to be assessed and would make it practical to obtain quantitative answers to complex questions for both general acquisition policy and specific procurements.

g. Finally, the Model could be used as a research tool for investigating developmental alternatives and managerial strategies. It could forecast, for example, the impact of different manning assignments, more or fewer development support facilities, longer or shorter schedules, etc.

## 1.2 PHASED DEVELOPMENT APPROACH

An early objective of the work on this project was to establish a balance between the scope of the work and the resources that could be made available for its accomplishment. The potential width of the Model was seen as the entire Major System acquisition process from the Conceptual Phase through the Deployment Phase. The potential depth of the work was limited only by what could practically be described, estimated, and ultimately simulated.

The achievement of the Model's full potential was taken as the objective of a multi-year effort. Scope limitations were necessary, therefore, to bound

the initial phase of the work. Some balance between width and depth needed to be struck.

While it is a functional necessity for definition to precede implementation, it is a practical necessity that the consequences associated with various implementation techniques be allowed to influence the style and content of the definition work. In order to achieve adequate depth, it became necessary to restrict the width. Since the width potentially includes all five Acquisition Life Cycle phases, the selection of at least one of these phases for the initial work was indicated. After some evaluation, the Full-Scale Development (FSD) Phase emerged as a logical choice for the following reasons:

a. Full-Scale Development on a system does not ordinarily begin until requirements have been established and an implementation concept has been selected and validated. This prior work provides much of the basis for sizing the software, and is accordingly an essential prerequisite to accurate costing. Thus, more accurate estimates could be made at the start of the FSD Phase than for earlier phases of the Acquisition Life Cycle.

b. The decision to embark on Full-Scale Development involves a high cost commitment in which software cost and timeliness are important considerations. Therefore, since the need for better estimates is most acute at this point, a FSD model seems most likely to find an early pilot application on an ESD project.

c. Since FSD is the life cycle period during which the expenditure rate is highest, a FSD model should achieve the highest payoff to ESD.

d. Finally, since the staff assigned to this project were most experienced with the FSD phase, they were better able to efficiently model that phase than any other during the critical Model formulation period.

It was also necessary to limit the depth of the project during FY 79. This was done by establishing an evolutionary development approach in which the first step would be basic core capability. This does not mean a simplistic "show something" product; it means, instead, one that provides for meaningful operation at a basic level, and is designed to accommodate the anticipated future growth.

To guide this growth, several successive versions of the Model were defined, and a plan for their phased development, pilot testing, and operational use was outlined. To help implement this plan, we were fortunate to devise a very powerful and flexible Simulator design concept, which will assure a relatively small computer program, easy to maintain and modify.

### 1.3 ORGANIZATION

This report has been organized into a report body that is supported by a number of appendices. The appendices generally serve to retain the documentation developed during the ongoing definition and design activities.

They incorporate the main technical substance of the work performed to date. While these appendices are too detailed for inclusion in the report body, they do provide important reference and backup materials that have not been distributed previously. For example, Appendix A, Process Flow Diagrams and Amplification Notes, especially Figure A-2, Software Acquisition Process Flow Diagram - LoSim Level, depicts the entire FSD Process, at the level planned for initial simulation. This figure may be inspected to obtain considerable insight into this Acquisition Life Cycle phase. While most readers are probably familiar with (or have participated in) the FSD process, the diagram can impart an integrated view of the whole procedure. The overall complexity and degree of interaction of the process are not so apparent when it is experienced during the two or three years during which it normally unfolds.

Similarly, Appendix B, Model Definition Data, contains estimates of the manpower and elapsed times necessary to complete each of the activities depicted in Figure A-2, and the probabilities of the decisions shown there. Appendix B may also be of interest because it represents the Figure A-2 flow diagram network in tabular form for use by the Simulator.

The appendices further show much of the progress achieved during the year and the degree to which the design has matured.

The report body summarizes and ties together the technical information contained in the appendices. For those persons whose interest in this work is casual, a summary is provided in Section 2. Section 3 describes the overall development approach. Sections 4, 5, and 6 describe, respectively, the techniques used to define, quantify, and simulate the process. Section 7 summarizes the accomplishments during FY 79, and the status of the project at the end of that fiscal year. Section 8 describes the areas of anticipated growth, and a plan for its achievement. Conclusions and recommendations are provided in Section 9.

The report contains numerous figures and tables. Each of these is located at the end of the section to which it is most pertinent. References to figures and tables in the body of the report which are not part of a section include page numbers to facilitate access. The List of Illustrations can be used to locate all figures and the List of Tables all tables.

## SECTION 2

### SUMMARY

Virtually all major military systems now include computers and associated software as essential elements for providing system functions. As these systems have evolved, the computer programs (termed embedded software) have grown in capability and size. They have also contributed greatly to system development costs and, all too often, to cost overruns and slipped schedules.

For these reasons, project planners need a method of obtaining reliable estimates for the cost and time of acquiring embedded software. While many methods are in current use, none have produced results with sufficient accuracy and reliability to meet needs.

Recently, a concept evolved which suggested that more accurate estimates could be obtained by using a simulator to model the process by which software is developed and acquired. By decomposing the overall process into unit functions which could be grasped and evaluated, and by allowing development decisions (and resulting rework) to be explicitly included, more accurate results, plus expected variations, would be obtained.

A project based on this concept was funded. The results of the FY 79 work are described in this report. The software-related work in the Full-Scale Development Phase of the Major System acquisition process has been carefully defined by means of several levels of process flow diagrams plus supplemental information. The process has been quantified to express the cost and time contributions of each unit process as well as the probability value associated with each decision element. A simulation concept was developed by which the whole acquisition process could be converted into tabular form and then used to drive a rather small computer program that conducts the simulation and collects the results. The actual Simulator design is well along, and some of it has been coded and compiled.

Considerable progress was made on a first model of the Simulator (Model 0). Future plans call for a series of successive models, each of which captures more of the inherent subtlety of the acquisition process. Also the later models will become more generic and therefore more readily adapted to a wide range of projects. Plans also include an increase in breadth of the Model so that all phases of the Major System acquisition process, from the Conceptual to the Deployment Phases, are included.

During FY 79, much has been accomplished and much learned. Based on that experience the Process Model looks entirely feasible and the original promise remains bright. Moreover, a number of other likely uses for the Simulator have been noted; its growth into a general management tool for wide use in acquiring embedded software appears promising.

Due to the results accomplished, and the potential value of a Software Acquisition Process Model Simulator, MITRE recommends continuing the project.

## SECTION 3

### APPROACH

The overall technical effort on the project is being channeled into three principal areas. These work areas, which are briefly introduced below, are detailed in Sections 4 through 6, respectively.

a. **Process Definition.** This work involves the creation of Process Flow Diagrams and Explanatory Notes which represent the process whereby computer software is acquired by ESD under the AFR 800-series regulations. In particular, the project is focused on software which is embedded within a large command, control, and communications system (see DoDD 5000.29). For maximum realism these Process Flow Diagrams represent both sequential and concurrent activities. Otherwise, to facilitate communication, they are conventional Von Neumann flowcharts.

b. **Process Quantification.** This work involves establishing parameters which describe each element within the Model, and obtaining appropriate initial values for these parameters.

c. **Process Simulation.** This task requires that the Model be mechanized so that it can be used to carry on synthetically the processes defined, using the assigned parameters. It can thereby forecast and report the statistical consequences in terms of probable schedule and cost distributions. A discrete event simulation program (i.e., the Simulator) is the mechanism of choice.

The Process Definition and Process Quantification work is based principally on the authors' personal acquisition program experience. This has been supplemented by review of pertinent regulations, specifications, standards, and data item descriptions (DIDs) performed to prepare a series of software acquisition management guidebooks.

The three work areas needed to be started in the order presented. However, because of the interdependence among them, the work was programmed to allow for the changes induced in each task by the "ripples" created by the others. For this reason, Process Definition was stressed at first until there was a basis for beginning the work in the other areas. Later, work proceeded simultaneously and interactively in all three areas, but with emphasis gradually shifting to simulation program design.

Another consideration in this conceptual development concerned the level of detail to be included in the Model and the extent to which the various Model processes would interact. After some exploratory investigation, an evolutionary approach was selected in which a relatively simple version of the Model would be defined and implemented first; this would be followed in turn by a sequence of increasingly realistic versions. This evolutionary approach led to a consideration of Model attributes, their utility, and the amount of effort needed for their definition and simulation. There was early recognition that an overall lifelike simulation of the process would be

complex and would need to include many subtle interactions. At the same time, it appeared that simpler versions could be useful, if built for limited applications. As a result, an initial version of the Simulator (i.e., Model 0) was defined and capabilities were established for several successive versions. In addition, other Model attributes were identified which could be added later, in response to the needs and priorities of the Model users.

Despite the desire for simplicity, the initial Model includes capability for dealing with basic design and acquisition concepts which are only indirectly treated (if at all) by other tools. One of these concepts is phased implementation (see Section 4.1.5). By using this concept the system software developer does not need to confront, design, and implement the required capability in its defined totality. Instead, he establishes a set of versions which are to be developed sequentially until the first deliverable version is completed. This concept was used with considerable success on a recent ESD project called SALTY NET, and is expected to receive widespread application in the future, particularly on major weapon system acquisition projects. This concept is also being used informally on this small-scale project.

Interaction and iteration are also explicitly represented in this Model. While these have always plagued the ongoing development process, other planning and estimation methods generally treat them indirectly (e.g., via loading factors) or ignore them. Since we believe that these factors are often largely responsible for the wide disparity between the project plan and its realization, the Model includes paths for both forward and iterative progression.

In addition, the Model includes decision elements which select among exclusive alternative paths. The Model also includes a mechanism for allowing the results of earlier processes to influence the consequent results of later ones. While this mechanism (i.e., the Special Event) will receive limited use in the earlier Simulator versions, it provides a basis for emulating much of the subtlety of the acquisition process. Later versions are expected to make extensive use of this capability.

Note that the provisions for phased implementation, interaction, decision elements, and iteration have been costly in terms of definition and design effort. They have been included, nevertheless, to avoid a design which could not accommodate the maturing needs of the Model users.

In order for a simulator to work, it is necessary that quantifying parameters be selected and specific parameter values be applied to the individual function boxes which populate the Model. Eventually, this will need to be accomplished by the development of valid relationships between the time expended in each function box and such factors as:

- a. the quantity and quality of manpower and other resources available;
- b. the extent and difficulty of the task represented by the box; and

- c. the size and complexity of the total project and of the computer program component or module being addressed by the box.

The development of these time and cost relationships and their validation require the expenditure of much effort and time - much more than have been available on this project. Since little prior work has been published (other than gross estimates such as those which ascribe cost distribution, like 40% design, 20% coding, and 40% test), this type of work would need to start from basics.

For these reasons, a more limited approach was taken for obtaining parameter values. In this approach, a set of parameter values (per function box) were developed for a typical project. These values were originally obtained from estimates made by persons familiar with the individual processes being modeled. The estimates were then calibrated by comparing the project schedule derived from these estimates with a typical schedule for such a project, as obtained from military procurement experience. The original parameter estimates for each segment of the schedule were then scaled to make the derived schedule coincide with the one estimated.



## SECTION 4

### PROCESS MODEL DEFINITION

In this section information is presented which describes our modeling of the software-related aspects of the Full-Scale Development Phase of the Major System acquisition process. Paragraph 4.1 describes the basic premises followed as the Process Model was being defined; Paragraph 4.2 introduces the set of diagrams by which the Model of the process was described. Paragraph 4.3 describes some prior projects wherein the authors gained relevant experience.

#### 4.1 BASIC PREMISES

During preparation of the Software Acquisition Process Model, it was found necessary to delineate the Model and to limit the scope of the effort to fit within a limited budget and schedule. The set of basic premises discussed below was established, therefore, as guidance for the initial phases of this work. Some of these apply to the acquisition process itself, others to simplifications introduced for application to early versions of the Simulator. These premises, whenever applicable, are referenced by Table A-2, Process Flow Diagram Amplification Notes, which supports the Process Flow Diagrams.

##### 4.1.1 Conformance to Military Standards

The acquisition process modeled will conform to all military standards and regulations that are normally applied to software acquired during Electronic System procurements. These include MIL-STD-483, Configuration Management Practices for Equipment, Munitions, and Computer Programs; MIL-STD-1521A, Technical Reviews and Audits for Systems, Equipment, and Computer Programs; AFR 800-2, Acquisition Program Management; and AFR 800-14, Vol. II, Acquisition and Support Procedures for Computer Resources in Systems. If deviation from these practices is found to be necessary, it will be explicitly described (and explained) at each point in the process where it occurs; a summary list of all such deviations, if any, will be provided.

##### 4.1.2 System, Segment, and CPCI Relationships

The relationships among activities associated principally with a system, its segments, and its Computer Program Configuration Items (CPCIs) will be considerably simplified in the early implementations. In particular, system segments can be used in different ways on different contracts and are therefore not fully amenable to generic implementation. For this reason, the Model addresses the CPCI (level 3) and one level higher. While this higher level is referred to as "system" (level 1) it could as readily represent "system segment" (level 2). The choice is dependent on the nature of the system and the specific contract(s) being simulated.

In addition, while the Model is designed to accommodate a number of CPCIs, it will treat these initially in a somewhat simplified manner. As thus modeled, all CPCIs will initiate and terminate together (e.g., in the System Test), and proceed independently in between. In actual practice, the various CPCIs often have dependency relationships which can be of critical importance to the success of a project (see Figure 6, notes B-D, page 39). Later versions of the Model will be designed to accommodate these relationships.

#### 4.1.3 Validation Phase Activities

The Process Model of the Full-Scale Development Phase presumes that a full Validation Phase has already been completed. However, since many projects omit this phase but incorporate some of its activities in the Full-Scale Development Phase, provision should be made for such activities' incorporation (e.g., the preparation of development specifications) in the FSD Phase Model. Extension of the Model to the Validation Phase is planned for later implementation. The process flow developed for that phase will be designed so that selected activities can be readily moved into the Full-Scale Development Phase.

#### 4.1.4 Support Facilities

The Model will presume that the Test and Programming Support functions will each be provided by separate facilities. On some projects, such facilities may be shared (in whole or in part) to support both functions. The Model will be designed to reflect any combined use of these facilities.

While the current Model provides for accumulating the costs of operating and maintaining support facilities and for the impact resulting from their late availability, it does not include the effect of contention between facility users or the results of unscheduled down time. These latter capabilities will be added in later versions.

#### 4.1.5 Phased Implementation Provisions

Procurement regulations allow design reviews to be conducted on a single or on an incremental basis. The Model is being designed to represent the incremental approach. While this decision adds to the complexity of the Model, it was taken because the single design review approach would not support the trend toward phased development, particularly for larger systems. The Model will also accommodate the single design review approach, simply by setting the number of design increments to one.

The initial Model is being designed to accommodate the following incremental approach:

a. Each CPI is defined by a specification which states the functional requirements to be met at the completion of the current procurement contract. While certain follow-on requirements may also be explicitly or implicitly defined, these are treated as beyond the scope of that contract.

b. The contractor would divide the total contractual requirements into several increments (hereafter called Developmental Integration Groups (DIGs)). This division would be defined in a phased implementation plan that is included within the Computer Program Development Plan (CPDP).

c. As shown in Figure 5, Phased Group Development Example, the contractor would then proceed with the design of the first DIG (DIG-I). The work on this DIG would then pass successively through the various stages of the design process (including Preliminary Design Review (PDR) and Critical Design Review (CDR)), and through coding, debugging, integration, and contractor internal testing. The work might also be subject to Preliminary Qualification Testing (PQT), but not to Formal Qualification Testing (FQT).

d. The design and implementation of the other DIGs would proceed in order behind DIG-I. Work on the second DIG (DIG-II) would begin after PDR on DIG-I; DIG-III would start after PDR on DIG-II, etc. Similarly, the CDRs and other development activities would proceed in the same order.

e. During each stage of development, each successive DIG would add to and build onto the aggregated preceding DIGs. In other words, a single CPCI would be built in successive stages; it would not be built as separate DIGs to be joined together at the end.

f. When the last DIG passed through each development stage, the total implementation to that stage would be complete. Therefore, each last DIG design review would be extended to survey the totality of the design, in addition to that of the last functional increment.

g. The Model documentation includes notation to accommodate the incremental development concept. The notation will indicate (with a "D"; see Figure A-1) those processes which are presumed developed in this phased manner. In addition, when a development phase is complete for one DIG, the process must return to that phase to begin work on the next DIG. This type of return is shown as type "D" on the Process Flow Diagram (Figure A-2) and in the Network Definition Table (Table B-1) (in its General Data Grp column).

h. The formal test activities may also be conducted on a similarly phased basis. The Model will support this approach by allowing Test Integration Groups (TIGs) to be sequentially processed in a manner analogous to the handling of DIGs. Note that the TIG division involves the test related activities and applies to a totally implemented CPCI; therefore, TIGs are not related to DIGs in terms of usage or quantities.

#### 4.1.6 Incidental Activities

While the Model is planned to include all significant mainstream acquisition activities, it will not include a number of incidental tasks that are essential to a project but that would add needless complexity to the Model. Instead, the cost and loading impact of such activities will be included as general overhead factors. Similarly, certain events and activities judged too infrequent or too inconsequential to the Model (though not to the acquisition process) will not be included. Should experience or

collected data indicate that some of these incidental activities be added to the Model, this can be done in a later version.

#### 4.1.7 Resource Utilization

Each process activity consumes project resources such as:

- a. contractor manpower in various job categories;
- b. government manpower in various job categories;
- c. development support facilities;
- d. test support facilities;
- e. miscellaneous other resources.

In the early implementations, only manpower resources are being assigned to specific process activities. There are two reasons for this. First, manpower is by far the most important resource in software acquisition. Second, because of this, we deemed it more important to develop reasonably sound initial manpower estimates than to divert effort toward estimating other resource requirements. The manpower categories listed below were selected for implementation based on our acquisition program experience. In addition, the manpower accounting techniques and the effects of resource limitation are described below.

a. Contractor personnel. Five job categories were selected for individual assignment to each activity:

- (1) systems engineer or analyst;
- (2) designer;
- (3) programmer;
- (4) test engineer;
- (5) support (e.g., equipment operator, librarian, documenter).

During our initial Process Model work, Management was included as an additional category. This separate Management category was abandoned when the need to subdivide a manager's time among many ongoing tasks made its estimation impracticable. Aside from the difficulty in estimation, results would be inaccurate because management styles differ widely and would generally be unknown. For this reason, we decided to represent management as a continuous activity with an overall manpower profile that conforms to the estimated (or given) needs for the project being modeled.

b. Government personnel. Three job classifications were selected for personnel assignment to specific activities; these reflect the three principal commands involved in system acquisition: The Developing Command (e.g.,

Electronic Systems Division (ESD)), the Using Command (e.g., Tactical Air Command (TAC)), and the Supporting Command (e.g., Air Force Logistics Command (AFLC)). Consideration was given to further specifying the assignments (e.g., to Engineering, Test, Configuration Management, etc.) but it was decided that this would be more appropriate in later versions.

c. Initial implementation technique. Recognizing that personnel categories are likely to differ for different contractors and projects, a generic assignment technique was dictated. The method selected can be used for any number of categories. While a one-dimensional list of manpower categories is being implemented, it can be expanded to two (or more) dimensions if needed. This would allow, for example, the group of ESD personnel assigned to an acquisition program to be further divided into functional groups, such as Engineering or Test. It would also allow contractor designers, for example, to be further distinguished as senior or junior, etc. Working versions of the Simulator will eventually use job categories that are compatible with those developed for the planned ESD Software Acquisition Resource Expenditure (SARE) reporting system.

d. Resource limitations. The rate of progress on any project can be strongly influenced by the quantity and quality of the available resources. The Simulator, in Model 1 and later, will allow the amount of resources to be defined such that each activity can draw from a resource bank when it is ready to begin. When the demand for a resource exceeds its supply, the process will slow accordingly. This process behavior, while inherently simple, may require that different and perhaps complex management strategies be devised to resolve automatically the problem of allocating scarce resources among competing activities. For this reason, the initial Simulator version will not reflect the effects of resource limitation. Instead, it will allocate manpower only on the basis of the needs of each activity, and keep track of the amount used. The end result of a simulation run will include a statistical profile of the total amount of manpower used in each category versus time.

#### 4.2 PROCESS FLOW DIAGRAMS

Process Flow Diagrams have been used as the principal means for describing the process of acquiring embedded software. They were developed at several levels of detail, as follows:

- a. a global view of the whole process;
- b. a high simulation level (HiSim);
- c. a low simulation level (LoSim); and
- d. expanded views of LoSim boxes to show more elemental relationships as needed.

The conventions followed by these Process Flow Diagrams are described in Appendix A, Figure A-1, Flow Diagram Notation. Briefly, they define three types of basic element: (1) function boxes; (2) auxiliary elements (e.g.,

connectors); and (3) lines of flow. These conventions should be understood before the Process Flow Diagrams are reviewed.

The LoSim and Process Flow Expansion Diagrams (Figures A-2 and A-3, respectively) distinguish support activities from mainstream activities by representing the former in trapezoidal boxes while limiting the use of rectangular boxes to the latter. This distinction is not made in the HiSim and higher-level Process Flow Diagrams (Figures 1-3). There, rectangular boxes represent all activities.

#### 4.2.1 Global View

The sequential relationship among the phases that constitute the standard Major System acquisition process (defined in AFR 800-2, Acquisition Program Management) is shown in Figure 1, Major System Acquisition Process Normal Phases of Progression. The Full-Scale Development Phase is emphasized in this figure to indicate that it is the focus of the initial work being accomplished on this project. The principal groups of Full-Scale Development Phase activities and the most important lines of flow among them are shown in Figure 2, Software Acquisition Process Model Activity Flow Overview - Full-Scale Development Phase. Besides providing a global view, this figure aids access to two more detailed diagrams, termed simulation level diagrams, that the Simulator can mechanize:

a. The box numbers in Figure 2 refer to those used in Figure 3, Software Acquisition Process Flow Diagram - HiSim Level. Two of the Figure 2 box numbers (i.e., 66 and 96), which pertain to the Program Management Review (PMR) and ECP processes, respectively, are not shown in Figure 3. Inclusion of these processes was deemed inappropriate because both can interact with most of the other ongoing activities in a way which cannot be properly represented at the HiSim level.

b. The sheet number references given in each Figure 2 process flow activity group box refer to Figure A-2, Software Acquisition Process Flow Diagram - LoSim Level. The box numbers in Figure 2 also refer to the digit portion of the alphanumeric box identifiers shown in Figure A-2.

#### 4.2.2 Simulation Level Diagrams

Two simulation levels are planned to support different purposes to which the Simulator may be put. The two levels may be either used separately or intermixed. This will let detailed simulation results be obtained in selected areas while the remaining portions of the process are treated more generally. Also, the whole process or just a portion of it may be simulated.

Note that transition from low to high level modeling involves both coalescence of many boxes into a few and abridgement of the lines of flow (i.e., network linkage). While box coalescence is readily accomplished, the need to retain network continuity will require that mixed-mode simulation level transitions take place only at points where the lines of flow are compatible.

#### 4.2.2.1 The High-level Simulation (HiSim) Diagram

The HiSim diagram (Figure 3) views the Full-Scale Development Phase as a serial and parallel sequence of 32 composite activities. This diagram also shows the main lines of process flow that connect these activities. A simplified notation has been used to label those connectors that reflect dependencies between activities that fall in different main lines of flow. Thus, all connectors that lead to documentation activities have labels that begin with a "D". Similarly, test-related connector labels begin with "T". The beginning and ending Terminals are labeled "B" any "Y", respectively, in agreement with the notation used on all the other process flow diagrams.

While Figure 3 provides a modestly detailed overview of the FSD process, it is important to recognize that the following are not represented:

- a. decisions (i.e., exclusive alternatives);
- b. task iteration;
- c. Integration Group (i.e., DIG and TIG) progression;
- d. most distinctions between contractor and government activities.

Because of these omissions, pure HiSim simulation appears to have marginal value except for general project planning and estimation before further detail becomes available. If thus used, the HiSim parameter values should, to be realistic, represent the results of simulation at a lower level.

The HiSim Process Flow Diagram looks more promising as the initial basis for a mixed-mode simulation; (i.e., one that includes more detail selectively). This detail would come from the LoSim Process Flow Diagram (discussed next) or modifications thereof.

#### 4.2.2.2 The Low-level Simulation (LoSim) Diagram

The LoSim Process Flow Diagram (Figure A-2) uses approximately two hundred boxes on twelve pages to represent the overall process. The function of each box is described in abbreviated English, but box size limitations make it desirable to code some of the information via box shapes as well as in special fields. The key to connector and box number locations in Table A-1 will aid in following the flow and in finding boxes referenced in the tables of Appendix B.

The initial Simulator (Model 0) is planned to operate at the LoSim level, but will not include Engineering Change Proposal (ECP) processing, shown on Sheet 12 of Figure A-2. The diagram at this level is more systematic and somewhat less detailed than that shown as mid-level in our April 1979 interim report. While the LoSim level appears about right to represent lifelike process behavior, level adjustments, up or down, can be expected as the Simulator matures and begins to support the needs of its users. As shown in Figure 4, Model Box & Predecessor/Successor Counts, a total of 187 boxes are

used in this representation (not including the ECP Process); the figure also provides a numeric breakdown of box types, Integration Group assignments, etc.

Note that the complexity of the Model, in terms of the number of boxes and the number of interconnections, is not critical at this time. The Simulator design, described in Section 6, can readily support any reasonable degree of complexity. Higher complexity will add to the time and cost of simulation, and to the effort needed to establish the parameter values by which each process is quantified; it will not change the Simulator program, however, which is table-driven.

#### 4.2.3 Expanded Views and Amplification Notes

The LoSim level attempts to show valid lifelike behavior of the acquisition process while maintaining a manageable level of complexity. At its level, however, some of the activities portrayed may not be clearly enough described or differentiated. For this reason, additional explanatory material is provided in Appendix A. The Process Flow Expansion diagrams (Figure A-3) further subdivide selected LoSim box activities. The Process Flow Expansion diagrams depict more elemental and thus more easily understood activities; they also clarify the box-to-box flow. This material also provides a better functional basis for establishing parameter values to be applied to the parent LoSim boxes.

Note that any Process Flow Expansion can replace the equivalent LoSim box(es) in the Simulator input if the user wishes to explore certain aspects of process behavior at a lower level.

Appendix A also includes Amplification Notes that clarify certain activities or that provide relevant background material. These cover only a fraction of the total Model; their completion has been deferred to FY 80.

#### 4.3 PRIOR PROJECT EXPERIENCE BASE

The definition of the Software Acquisition Process as described herein is based mainly on experience obtained by the authors through work on prior software-related projects; the principal ones are listed below.

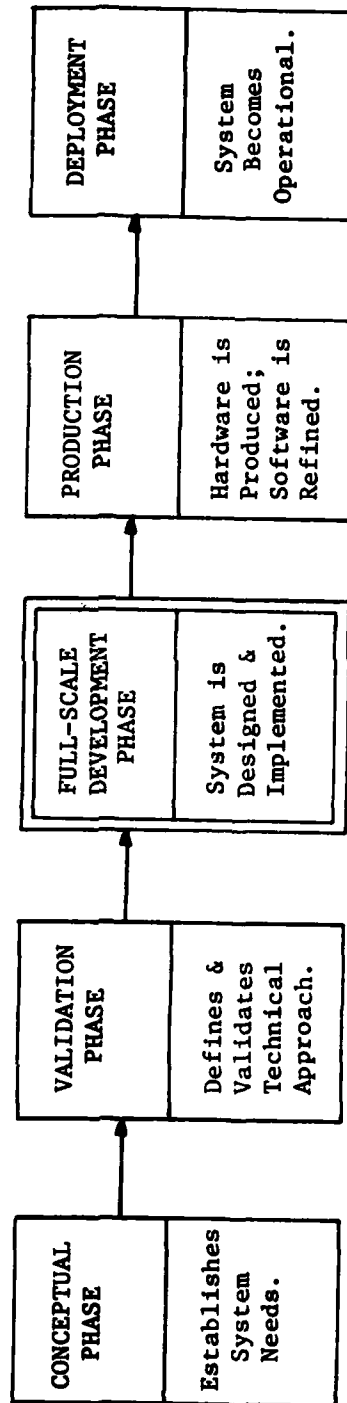
- a. Project 407L, Tactical Air Control System (TACS).
- b. Project 485L, Tactical Air Control System Improvements.
- c. Project 411L, E3-A (formerly AWACS).
- d. Project 427M, Norad Cheyenne Mountain Complex Improvement Program.
- e. SALTY NET III. This added a new capability to the existing TACS. This made it possible for the TACS to achieve physical and operational interoperability with NATO Units via the Link 1 Communication channels.



f. Project BARSTUR, an underwater tracking range system installed for the Navy near Hawaii. This project involved direct software implementation experience while working for a contractor (ITT Laboratories, Nutley, NJ).

g. Software Acquisition Management Guidebooks. This work involved extensive review of current Air Force standards, regulations, and practices that may be applied to embedded software acquisition. The intent of this project was to explain and clarify the software-related aspects of the acquisition process.

It should be observed that while this Model was built upon experience involving many projects over many years, these constitute but a small fraction of total Air Force acquisition experience. For this reason, changes in the Model should be expected as the experience base grows through constructive review by persons familiar with other system acquisitions and as the Model is applied to new acquisitions.



NOTE: Figures 2 and A-2 show increasingly detailed diagrams of this phase.

Figure 1. Major System Acquisition Process  
Normal Phases of Progression

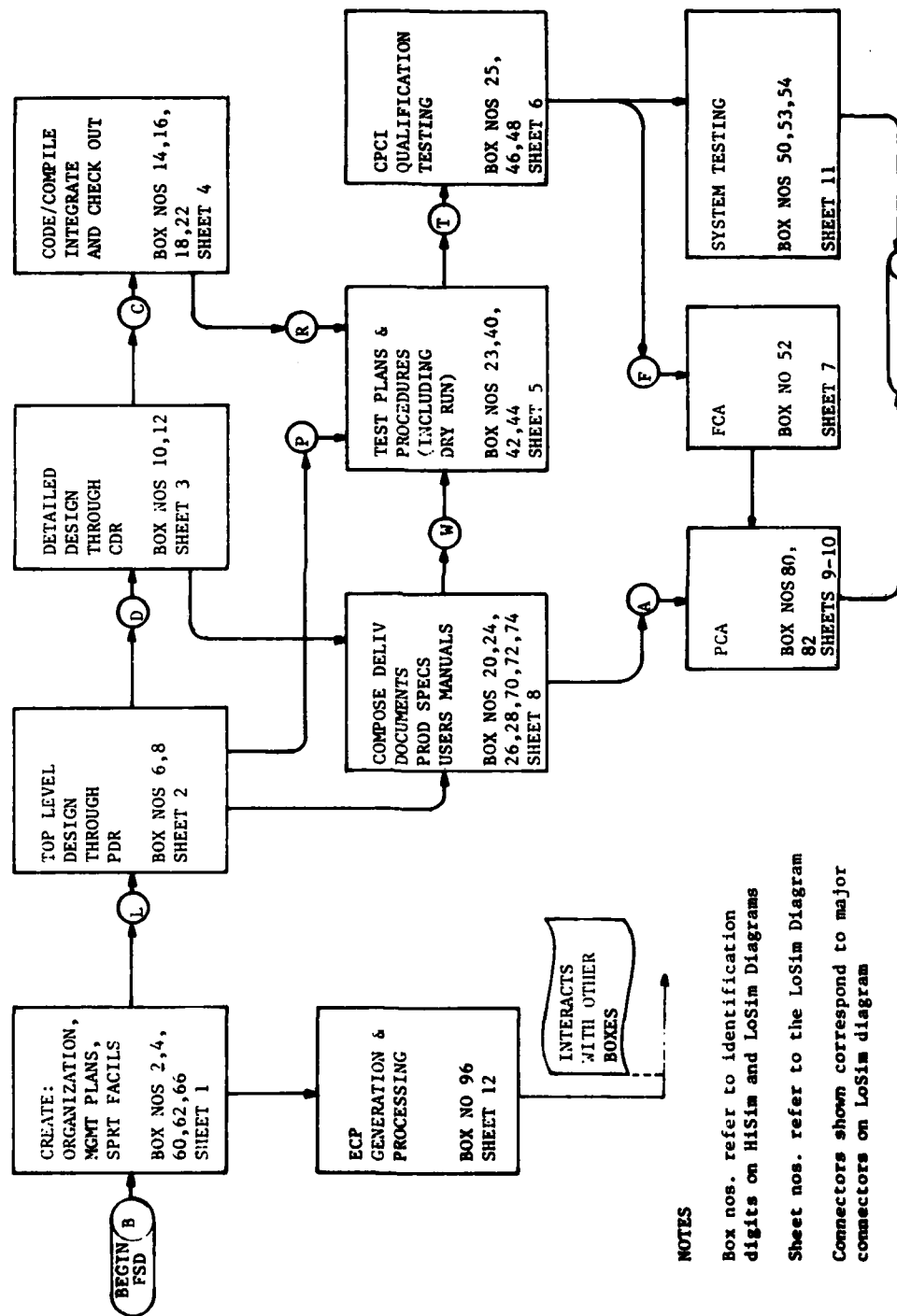


Figure 2. Software Acquisition Process Model Activity Flow Overview - Full-Scale Development Phase

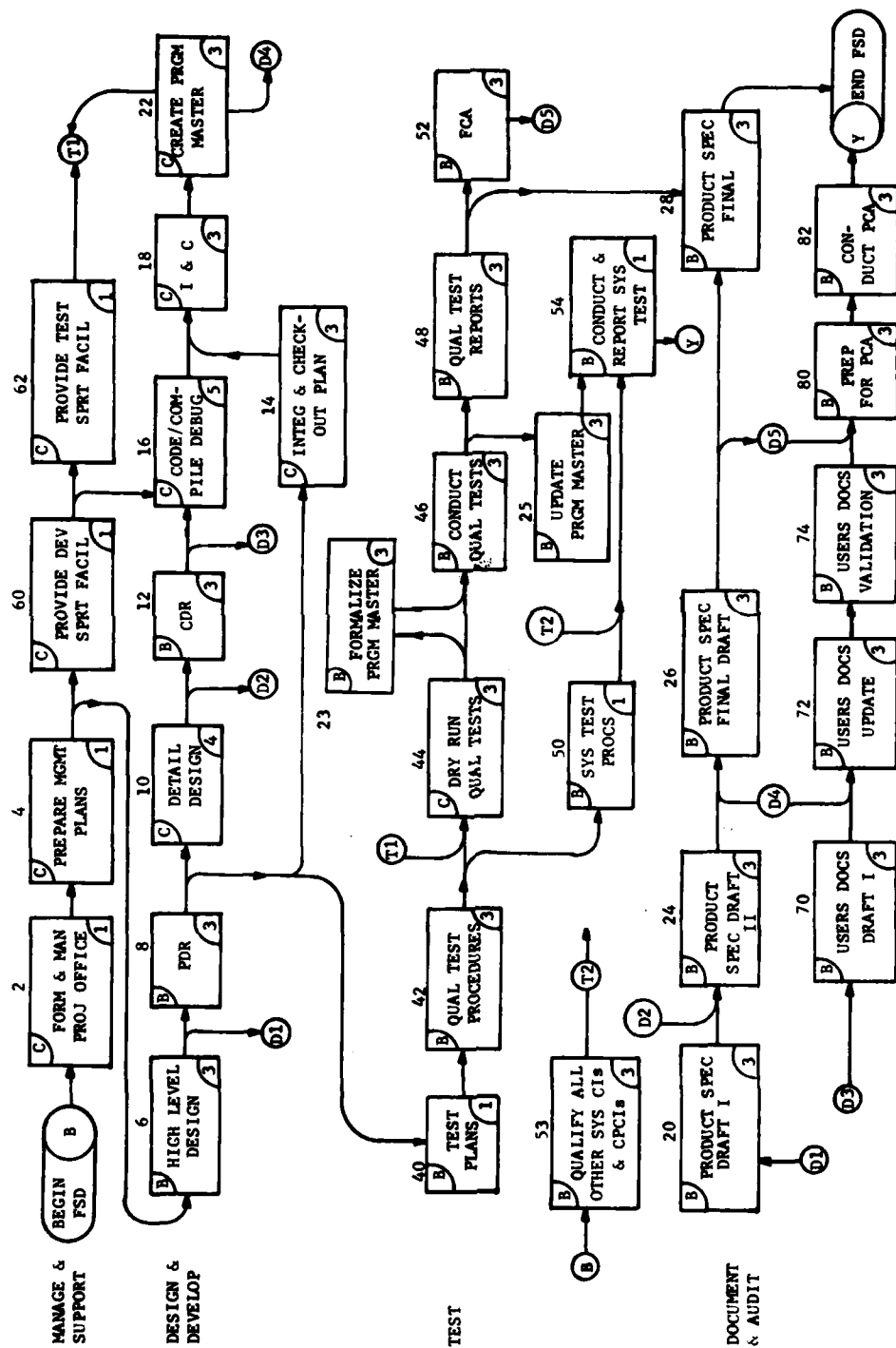


Figure 3. Software Acquisition Process Flow Diagram- HiSim Level

BOX TYPES	BOX COUNTS			TOTAL
	DIG	TIG	NEITHER DIG or TIG	
ACTIVITY	23	11	90	124
SUPPORT	1		4	5
DECISION	11	4	23	38
COUNTER	6	5		11
SPECIAL EVENT	<u>3</u>	<u>1</u>	<u>5</u>	<u>9</u>
TOTAL	44	21	122	187*

PRED(ECESSOR) COUNT - 284

NO.SUCC(ESSOR) COUNT - 57

YES.SUCC(ESSOR) COUNT - 245

\*Total number of boxes does not include ECP processing

Figure 4. Model Box & Predecessor/Successor Counts

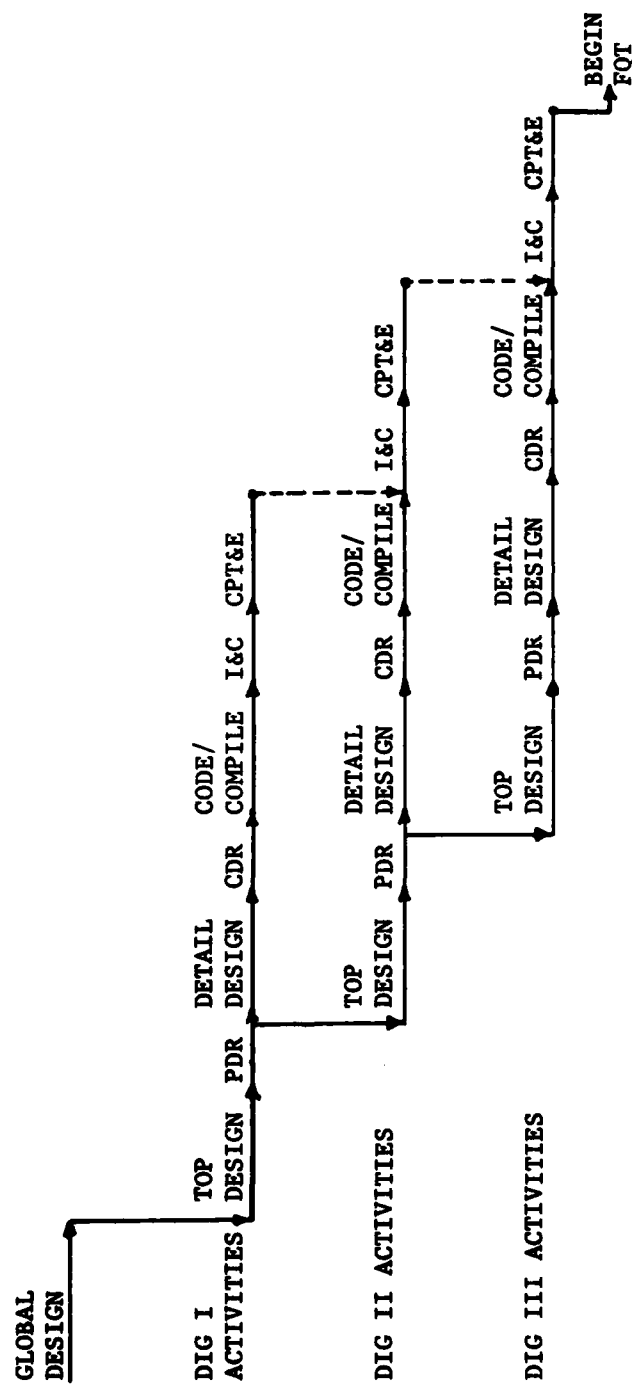


Figure 5. Phased Group Development Example

## SECTION 5

### PROCESS QUANTIFICATION

The Process Flow Diagrams and Amplification Notes discussed in Section 4 describe the sequences of activities and decisions involved in the acquisition (including development) of embedded software. Since this description is qualitative, it can yield no quantitative predictive output. In this section means are described for adding quantity and probability to the Process Model.

Ultimately, the parameters used and their relationships will be expressed generically so they can apply to a wide range of projects. That task is most difficult and will require data sources with much better definition and control than are now available. For example, the planned ESD Software Acquisition Resource Expenditure (SARE) reporting system will yield such data. For this year's effort, quantities have been developed to reflect those obtained on a typical Air Force system. This approach involved the steps listed below and described in the following paragraphs:

- a. definition of a typical system's software;
- b. selection of parameters to define each type of functional element contained in the Model;
- c. estimation of parameter values for each Model element;
- d. calibration and refinement of the estimated data.

#### 5.1 SOFTWARE DEFINITION FOR A TYPICAL SYSTEM

The size of a typical project's software generally falls between one hundred thousand and one million lines of code. For our work, an earlier project (with which the first author was familiar) was selected as a mid-range example. This project, 407L - Tactical Air Control System, was first developed about ten years ago (1967-1971) and is still being used. The data used herein do not correspond exactly with those of the original development. Instead, they have been modified to reflect changes in the evolving development process, including improvements in tools and techniques as well as increased training and deeper skills. In addition, all computer program sizes, time durations, and manpower levels have been smoothed and rounded.

Based on the above, the CPCI composition of a typical system was defined, as shown in Figure 6, Typical Project - CPCI Breakdown. The most significant and generally most critical of these CPCIs, (i.e., the real-time system Operating Program) was used to estimate the typical CPCI-level data for use with the Model.

A related milestone schedule is shown in Figure 7, Typical Milestone Schedule for Operating Program. Note that while 407L actually consumed 42 months, the 30 month schedule in Figure 7 reflects our assessment of the industry's current capability. Similarly, the corresponding contractor manning profile, shown in Figure 8, Typical Manpower Utilization Levels, is considerably lower than that expended during the actual project. Finally, although the original CPCI was not divided into Developmental Integration Groups (DIGs), our Model assumes a three-DIG division.

A set of manpower cost figures, based on projected manpower usage and reflective of current labor rates, is shown in Figure 9, Typical Manpower Costs for Operating Program CPCI. These figures, which should be viewed as rough approximations, are adequate to support their purpose in this study, which is to provide a basis for the calibration and refinement of the initial parameter value estimates. Consequently, they ignore the time value of money, which later Model versions will provide for. Also, Figures 8 & 9 combine (as SYS/TEST) systems engineers, analysts and test engineers, because often the same personnel perform these functions at different times during an acquisition program, and thus earn at comparable rates.

## 5.2 PARAMETER SELECTION

For each of the three basic box types represented in the LoSim and Process Flow Expansion diagrams, the following types of parameters have been selected.

### 5.2.1 Activity Box Parameters

Each activity consumes resources, including manpower and time, often expressed as man-days. However, since the number of man-days is strongly influenced by the manning level, it was decided that both an appropriate manning level and a duration would be assigned to each activity. We plan during FY 80 to estimate other needed resources, such as computer time, but have so far concentrated on manpower because it is normally the resource of overriding importance.

Note that the initial Model version represents the manning levels and durations independently; however, their estimates were developed jointly. Ultimately the Model will relate these and other resources explicitly, through parametric equations, which will also reflect explicitly the effects of software size and difficulty, development aids, management policies, etc.

A number of persons typically works on several concurrent activities on a split time basis. The Model allows, therefore, for fractional manning levels.

In addition, any activity may need to be repeated (possibly several times). Thus, iteration factors are provided to change the duration and manning levels appropriately for each successive iteration.



### 5.2.2 Decision Box Parameters

During the early development of the process flow logic, no restrictions were placed on the number of outcomes per Decision Box. As the logic matured it was observed that only a few decisions had more than two alternative exits. Since a uniform dual exit structure would simplify the design and implementation of the Simulator, and since dual exit boxes could be readily staged to reflect multiple decision results, Decision Boxes were restricted to two mutually exclusive exits. All decision questions were then phrased to allow answers to be uniformly expressed as "Yes" or "No". On this basis the probability of taking the "Yes" exit (pYES) was assigned to each Decision Box. (Since  $pNO = 1 - pYES$  it was unnecessary to assign "No" exit probabilities.) Since iteration can cause multiple reentry into a Decision Box, a corresponding set of decision probabilities was assigned to each iteration.

The assigned probabilities are typical of those observed on military procurements. Experience can vary widely, however, depending on factors such as contractor skill and experience, schedule and cash-flow pressure, government monitoring zeal, and urgency of need for the software or the system. Future Model versions will allow the probabilities to vary with the quantity and quality of the effort expended on the products on which the decision is based, as tempered by the expected contractual environment.

### 5.2.3 Special Event Box Parameters

The Special Event Box supports two different functions. One of these is to provide a label for important events, usually called Milestones, because a Milestone may not correspond exactly with any Activity Box or Decision Box.

The other Special Event Box function is to effect action at another point in the process (e.g., to increase the ECP frequency after each specification is delivered). While the Special Event Box is little used in the initial Process Model version, future versions will use it extensively to allow the computed quality of earlier activities to impact subsequent activities or decisions that depend upon the earlier work.

## 5.3 PARAMETER VALUE ESTIMATION

Specific estimates of typical parameter values were required for each Activity Box and Decision Box in the LoSim Process Flow Diagram. The Activity Box estimates were made by evaluating, based on personal experience, the work that must be accomplished per box and a reasonable manning level for its accomplishment. The Decision Box outcome probabilities were estimated similarly. These approximations need not be very accurate during early exercise of the Model. They can be improved by the calibration process described below. Appendix B describes and contains the parameter estimates made to date.

#### 5.4 PARAMETER VALUE CALIBRATION

Since no objective (i.e., measured) data are known to exist for most unit tasks (activities) and decisions defined in the Model, the initial parameter values were obtained as estimates from persons familiar with the overall process. While such values are suitable for Simulator development, the validity of the results obtained during its initial use will depend on the accuracy of the parameter values assigned. Even though Model uses such as sensitivity analyses can be helpful with roughly approximate data, the validity of and confidence in such results will be higher if the Model produces data which are consistent with normal experience.

Even though no objective data are available for most unit tasks, some cost and schedule data are available for the overall software acquisition process. While these data vary greatly, typical data can be chosen that are well within the range of common experience; the data described above are an example. The calibration process described below uses typical overall cost and schedule data to improve the initial estimates.

##### 5.4.1 Activity Box Timing Calibration

Calibration begins by bringing activity timing into conformance with a typical project schedule, such as that shown in Figure 7, as follows:

a. The original time estimates for the unit activities are summed in conformance with the sequences shown in the LoSim Process Flow Diagram (Figure A-2). This requires that the Activity Box timing data and Decision Box probability values be taken into account with both reflecting the effects of iteration. The details of this process are shown in Figure D-1, Activity Duration Calibration, and the data obtained are summarized in Figure D-2, Activity Duration Calibration Summary.

b. Using the data obtained in Step a, a most probable schedule is derived showing the principal acquisition Milestones.

c. The milestone timing obtained in Step b is compared to these same Milestones shown in a typical acquisition program schedule, derived as in Section 5.1.

d. A correction factor is then calculated for each inter-milestone time period. Each correction factor is then applied against all the Activity Box timing values within the corresponding inter-milestone period.

This process was applied to the main (critical) path of the overall process, covering the period from contract award through Formal Qualification Testing (FQT). The results, summarized in Figure 10, Activity Duration Calibration Summary - Main Path, show the timing after the calibration correction factors were applied, and after the average effect of iteration had been included. Figure D-2 gives the same summary data but shows the original estimated data with iteration and, also, the calibrated data without iteration. By comparing the figures it can be determined that multiplication by a factor of 0.75 was needed to bring the original estimates into

conformance with the typical schedule. The results also show that iteration increased the overall acquisition process duration by an average of 28%.

#### 5.4.2 Manpower Assignment Calibration

Once timing values are established for the main path activities, and the effect of iteration is taken into account, the estimated manpower assignments to each Activity Box can be calibrated as follows:

a. An overall timing diagram is drawn, as shown in Figure 11, Program Acquisition: Activity Sequence and Timing Diagram. Every activity is then placed on the diagram at a location conforming with its predecessors' iterated duration. This diagram thus shows what activities are likely to be going on at any time during the project.

b. When the manpower levels for each activity are each assigned to its appropriate time period, it becomes possible to sum the manpower level at any point in time.

c. When these summed manpower assignments are plotted vs. time, a manpower need profile is created for each category of personnel.

d. The manpower needs can then be compared with the typical manpower utilization levels shown in Figure 8.

e. The manpower levels assigned to each Activity Box can then be adjusted (and smoothed) for approximate conformance with the levels expected on the typical project.

The timing layout shown in Figure 11 preserves the functional and connective relationships among the activities developed on the LoSim Process Flow Diagram (Figure A-2). To facilitate reference to Figure A-2, the same connector identifications were used in Figure 11. Further work on the manpower assignment level calibration was deferred to FY 80, because Simulator design was deemed more urgent.

#### 5.5 FUTURE ACTIONS

Several types of action will be taken in FY 80 and later to improve the accuracy of the Process Model parameter estimates, and to relate these parameters explicitly to one another and to other factors believed to affect the elapsed time and costs of embedded software acquisition. The most promising of these types of action are listed below, roughly in order of increasing difficulty and remoteness. Each is then briefly discussed.

- a. review by a panel of experts;
- b. training course use and follow-up;
- c. sensitivity analysis using simulation; and

d. analysis of formally collected data.

5.5.1 Review by a Panel of Experts

The parameter values currently established represent the views of the authors. While we have had considerable experience with military project monitoring, as well as computer program development, the current data are more reflective of subjective evaluation and remembrance than of objective record. Such subjectively derived data can be improved by including the opinions of many experienced personnel. While such a panel would provide a basis for consensus values, it also would very likely show significant differences; these latter point to areas where further work should be concentrated. While the initial estimates should involve MITRE/ESD personnel, contractor employees with experience in military system development would also be valuable reviewers. These latter persons would be particularly qualified to provide estimates for activities normally performed by contractors.

5.5.2 Training Course Use and Follow-Up

One of the anticipated uses of the Process Model is for training military (and support) personnel for work on system acquisition. Such training would include instruction on the use of the Model for project forecasting and project monitoring. During the initial training classes, it will be necessary to stress the tentative nature of the initially used parameter values and the consequent need for tuning against real contract data. After a cadre of trained persons become involved in system acquisition, their reports on time and cost inconsistencies between the actual and forecast figures will become available. Cognizant personnel at the Model improvement and maintenance facility should use these reports as a basis for adjusting the parameter values (and cost estimating relationships) to improve the fit between Model data and actual results.

5.5.3 Sensitivity Analysis

The parameter value calibration procedure described in Section 5.4 was slow and arduous because it required manual tracing through the complex network, including Integration Groups and iteration loops, while time and manpower values were calculated and summed. Once the Simulator becomes available, the network tracing and computation will become automatic, fast, and accurate. This will allow the effects of parameter value changes to be reflected immediately into consequent results. By using this sensitivity analysis technique in this manner, the network can be tuned quickly to reflect actual experience on past (as well as current) projects.

In particular, when the typical project estimates used in the current Model are replaced by generic functions, sensitivity tests can support the verification of these functions and the tuning of the associated parameter values. This generic tuning and verification procedure is too complex to be performed manually. For this reason, quantification refinement by this method must be deferred until the automated Model becomes usable.

Once the Model comes into use as a contract monitoring aid, feedback data on the results of its use will allow the quantification of functions and parameters to be periodically improved and refined. At the same time, use of the Model will provide an early indication of trends in the quality of system acquisition on an industry wide basis. It will also provide a means for objectively comparing the quality of performance of different contractors, even on different projects. When this point is reached, meaningful performance incentives can be added to acquisition contracts.

#### 5.5.4 Analysis of Formally Collected Data

A separate ESD/MITRE activity, the SARE reporting project, is developing requirements for the uniform reporting of contractual performance data. When SARE is applied to future acquisition projects, the data collected can provide a basis for establishing the quantitative relationships needed for the Model. Because of the close relationship between the SARE project and the Process Model, compatible cost relationship structures are being planned.

CPCI	OVERALL SIZE			DIG-I		DIG-II		DIG-III		NOTES
	KMI*	CPCs	(DIGs)	KMI*	CPCs	KMI*	CPCs	KMI*	CPCs	
1. OPERATING PROGRAM (OP)**	80	18	3	40	8	20	5	20	5	A, B, C, D
2. SUPPORT UTILITIES (SU)	20	6	2	10	4	10	2			A, B
3. OFFLINE FUNCTIONS (OLF)	30	6	2	20	3	10	3			A, C
4. TEST SUPPORT FUNCTIONS (TSF)	25	5	2	10	3	15	2			A, D
5. HARDWARE DIAGNOSTICS	45	8	3	10	3	15	3	20	2	A
TOTALS	200	43	12	90	21	70	15	40	7	

- A. Approximate size divisions: 200 machine instructions per module; 25 modules per CPC.
- B. DIG-I of support utilities needed before operating program integration.
- C. Off line functions must be available for system test with operating program.
- D. TSF-I needed before FQT; TSF-II needed before system test.

\* KMI is machine instructions (thousands).

\*\* CPCI Quantification data developed for the model are based on just the operating program.

Figure 6. Typical Project - CPCI Breakdown

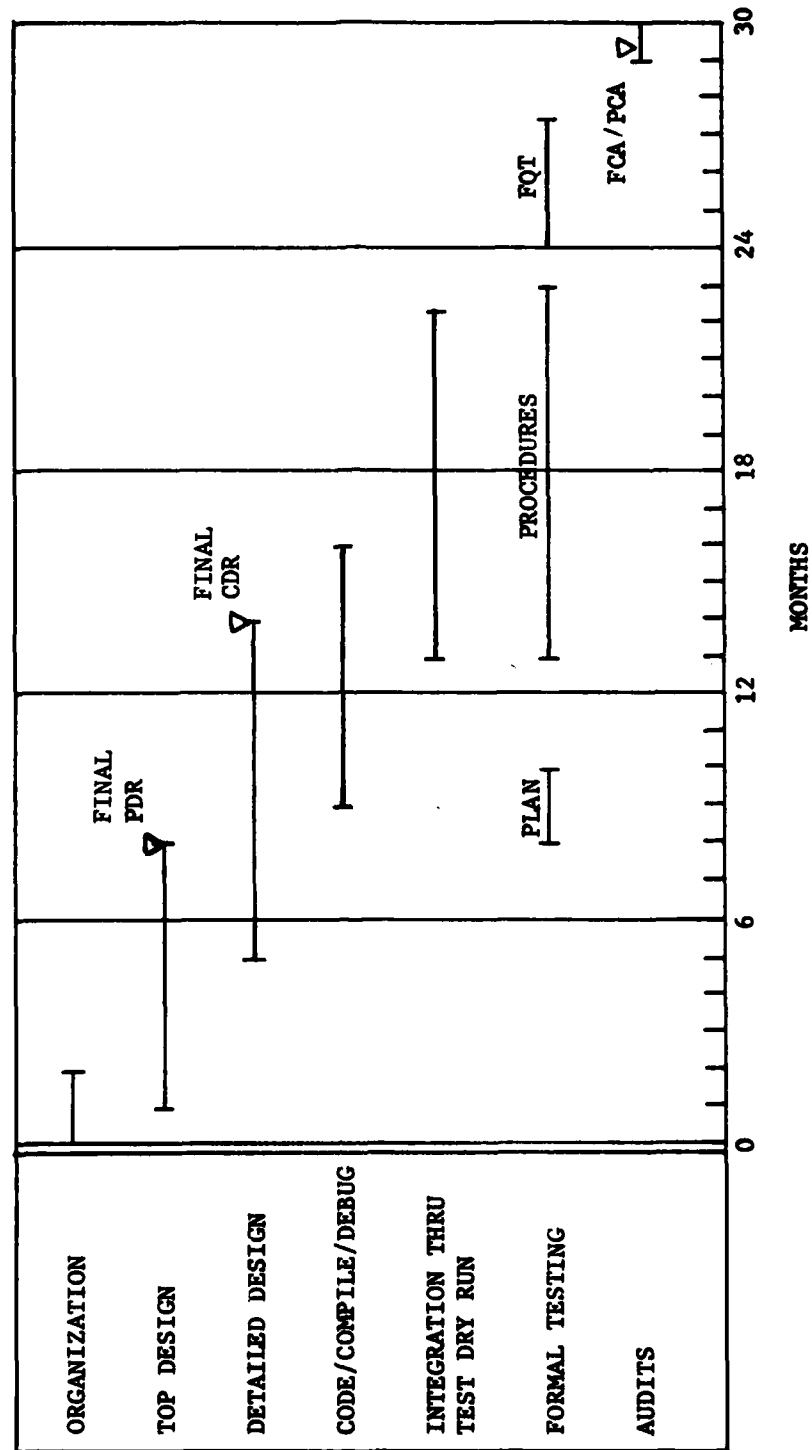


Figure 7. Typical Milestone Schedule For Operating Program

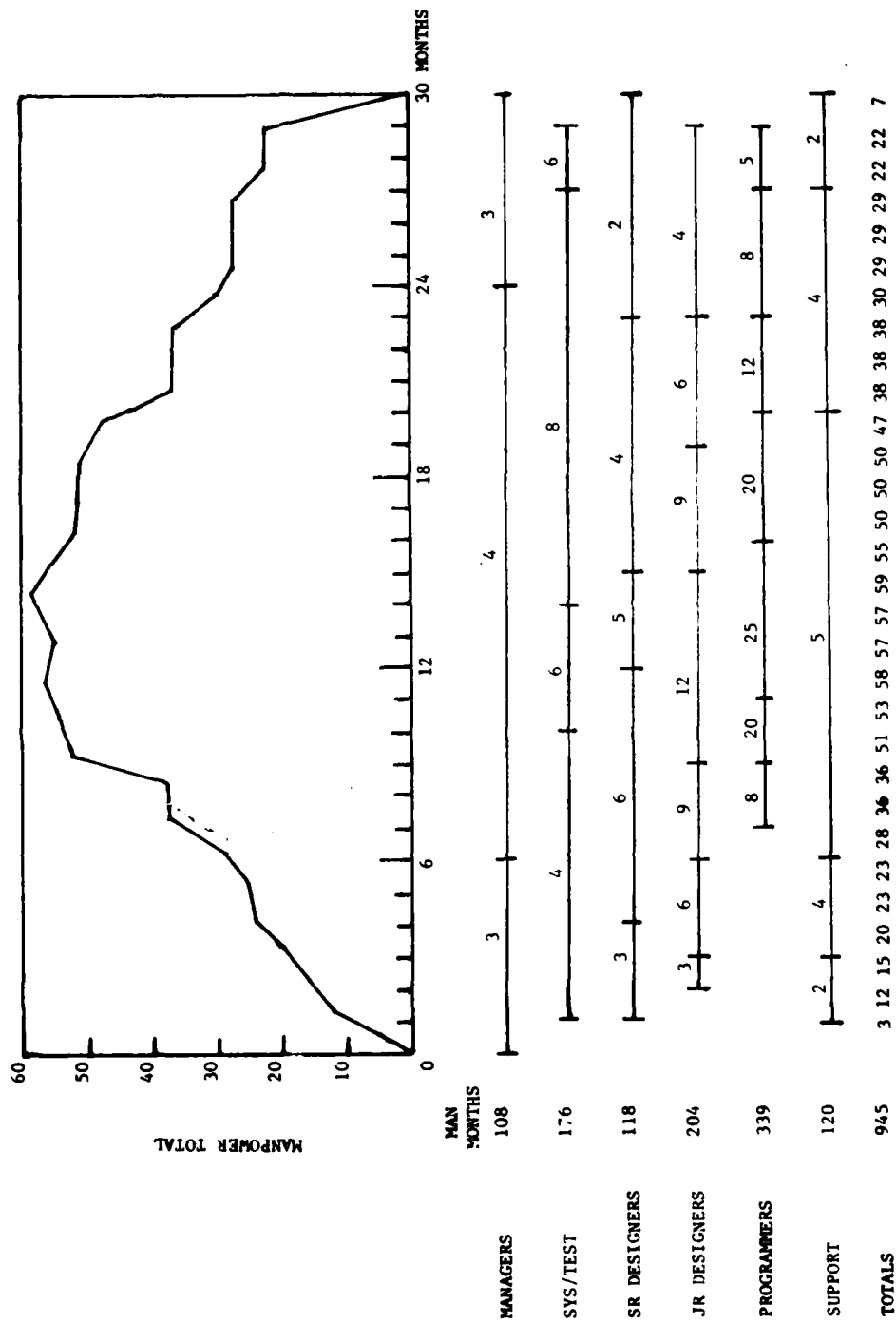


Figure 9 . Typical Manpower Utilization Levels



	MAN MONTHS	ASSUMED RATE*	COST (ROUNDED)
MANAGERS	108	\$ 6.6K	\$ 700K
SYS/TEST	176	5.3K	900K
SR DESIGNERS	118	5.7K	700K
JR DESIGNERS	204	4.8K	1000K
PROGRAMMERS	339	4.4K	1500K
SUPPORT	120	4.4K	500K
TOTAL	1065		\$ 5300K

\*Rates assume typical overhead of 120%.  
Costs for G&A, profit and support resources not included.

Figure 9 . Typical Manpower Costs For Operating Program CPI

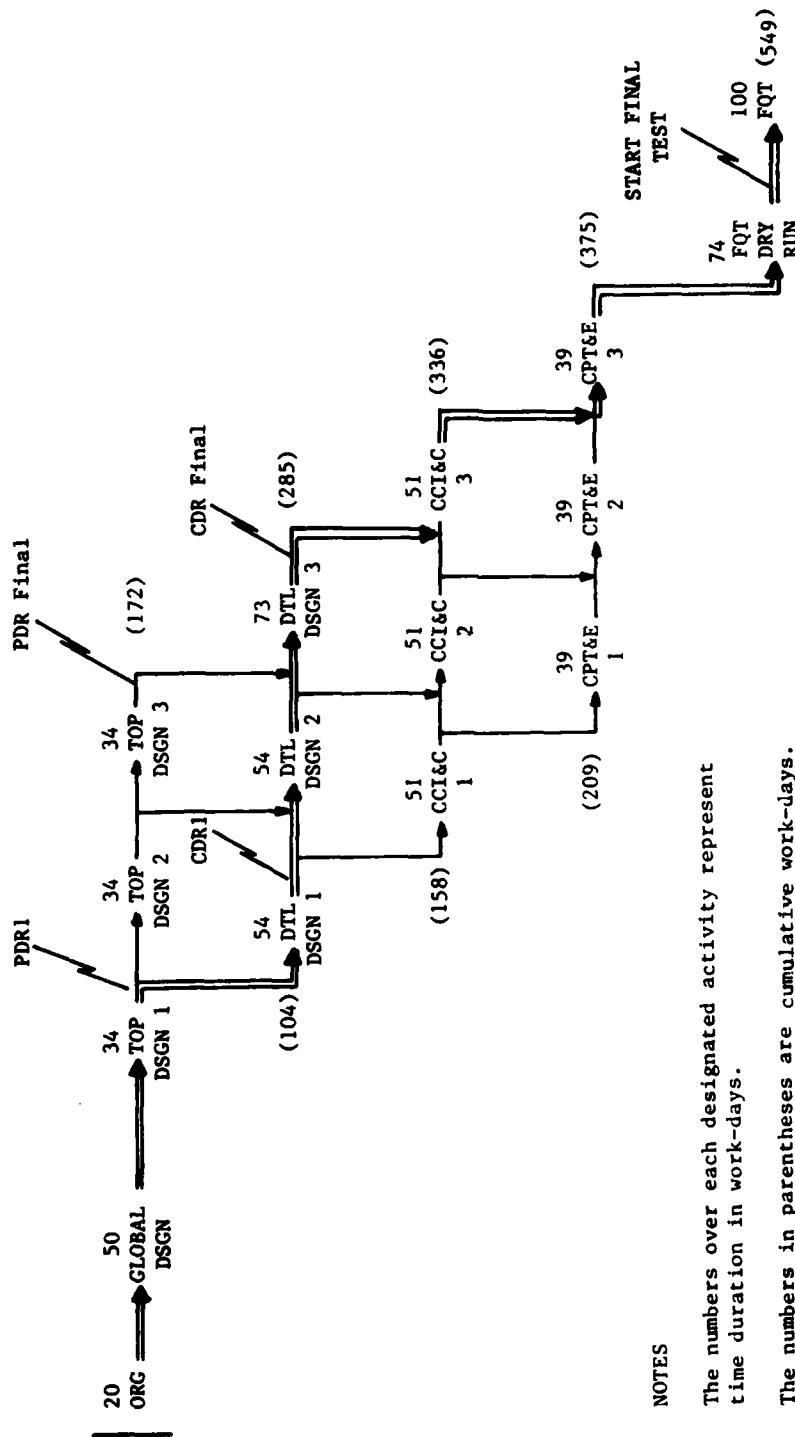


Figure 10. Activity Duration Calibration Summary - Main Path (Calibrated Activity Durations, With Iteration)

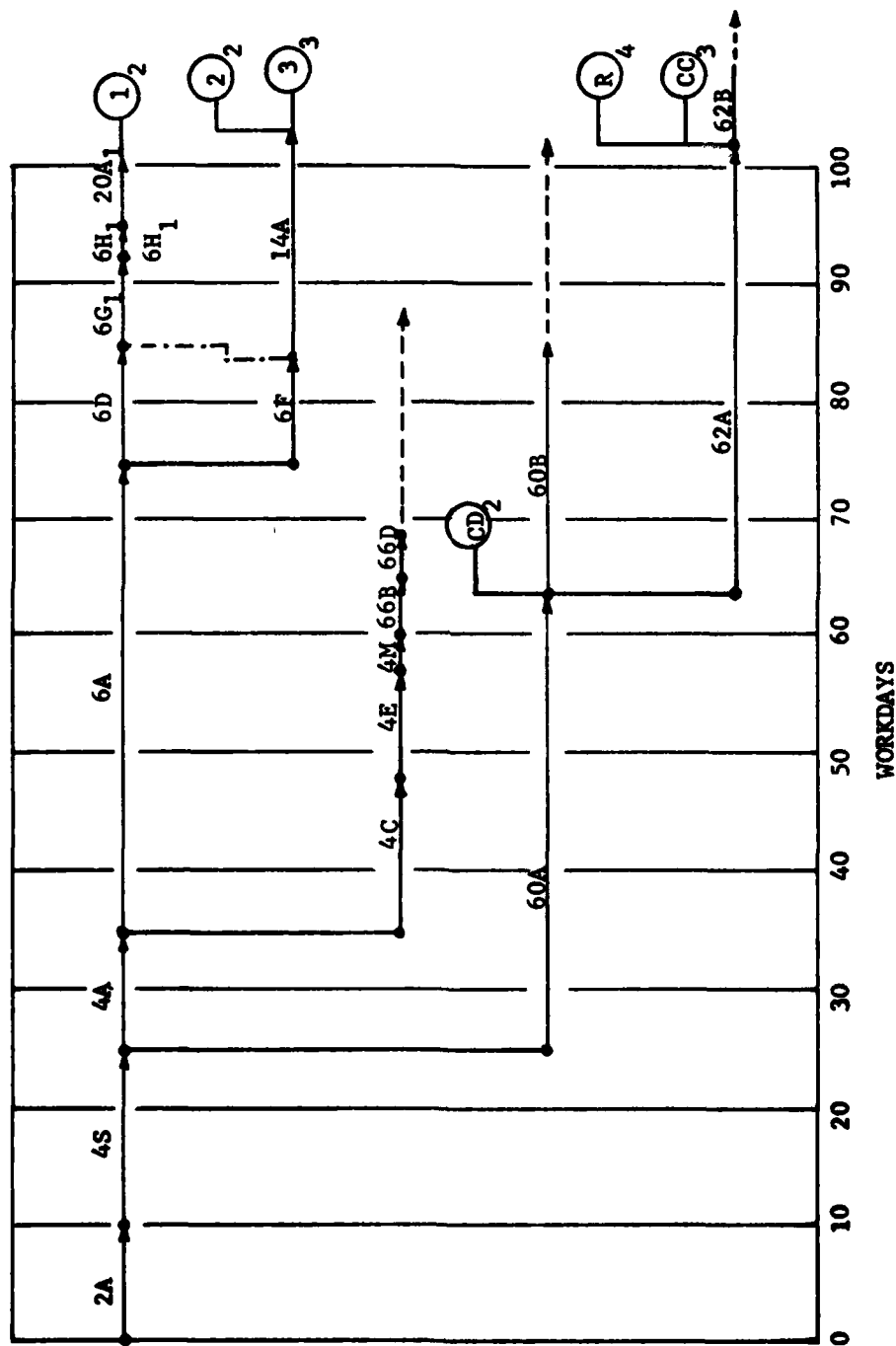


Figure 11 . Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 1

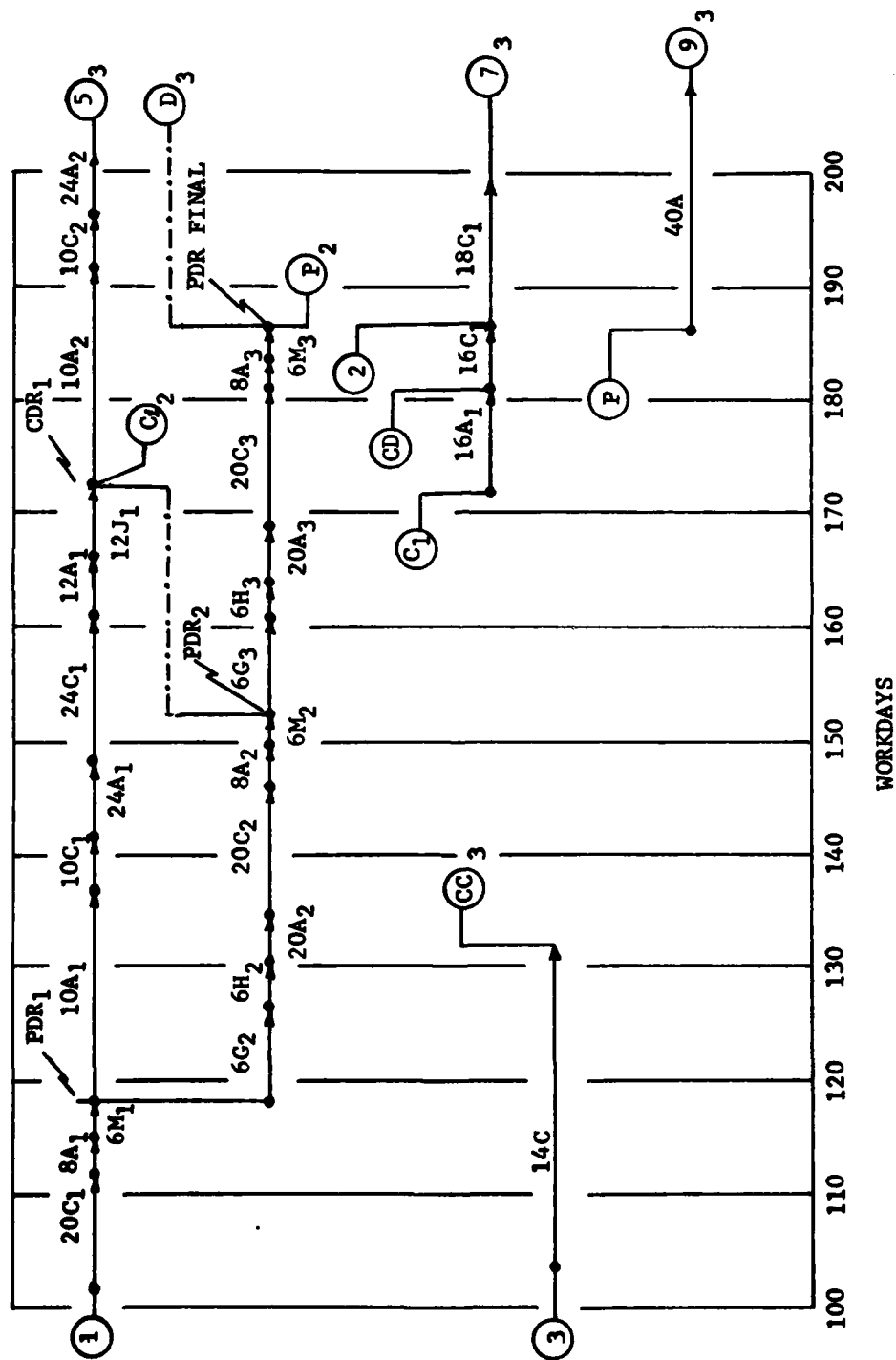


Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 2

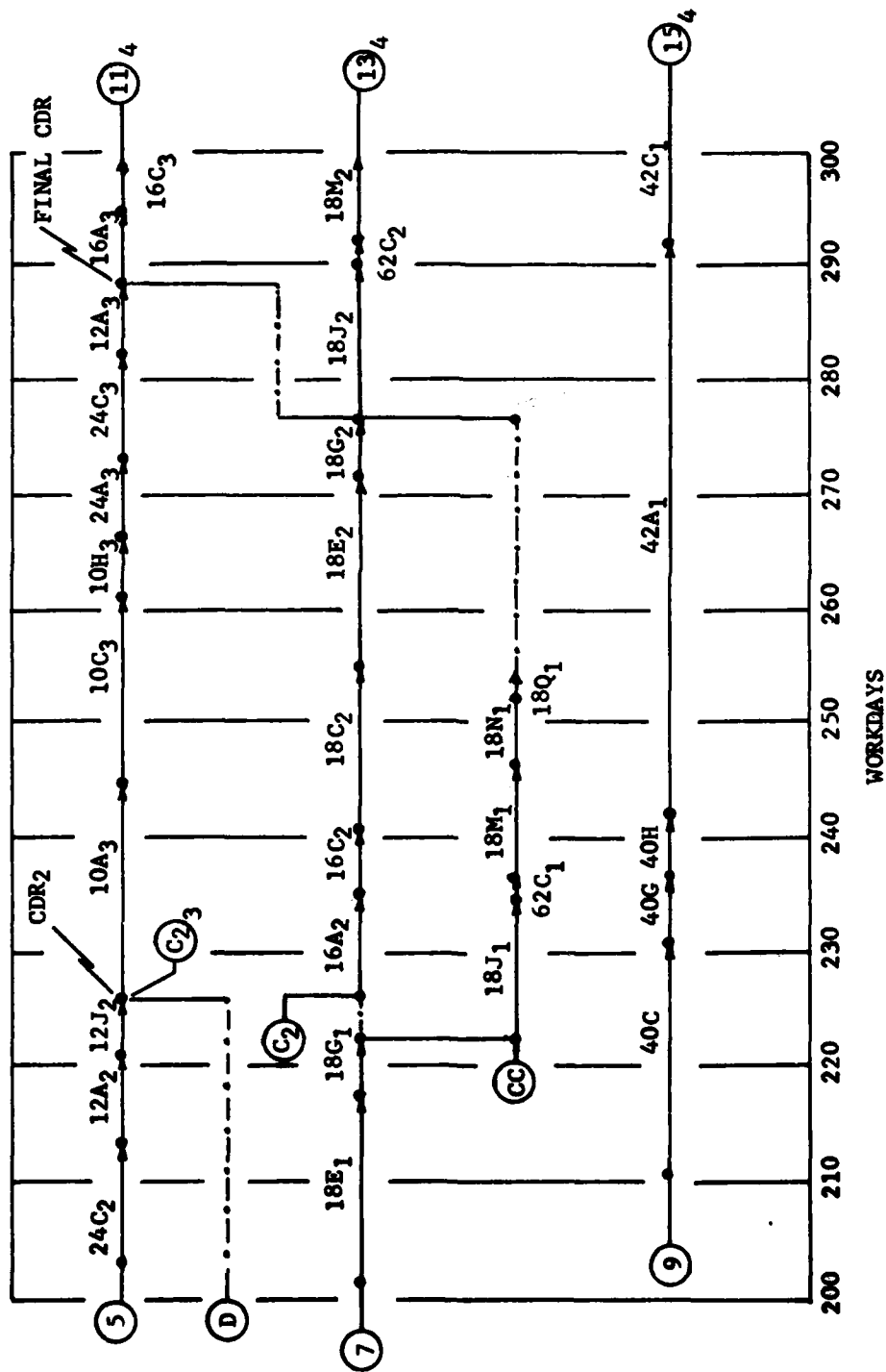


Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 3

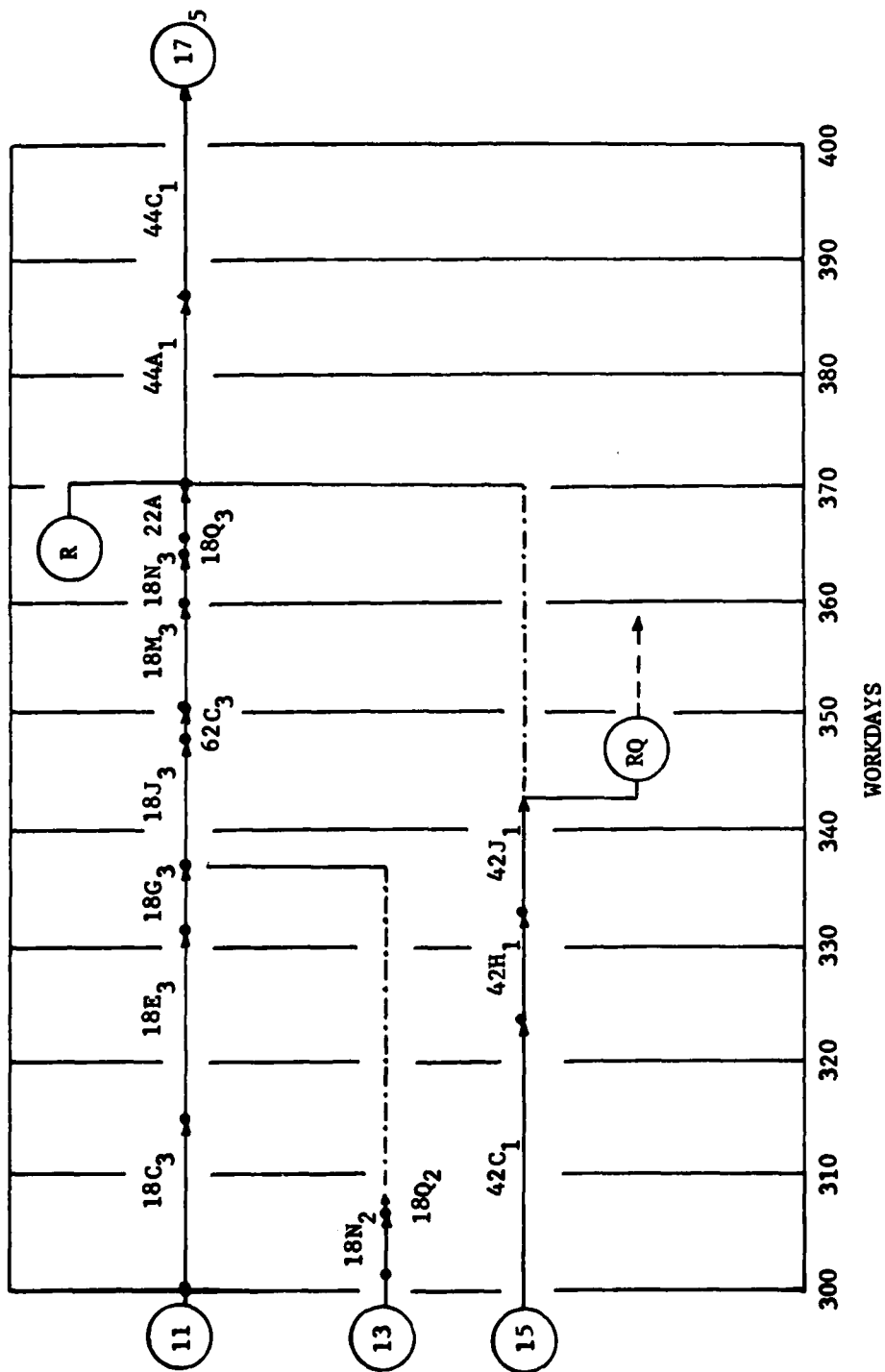


Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 4

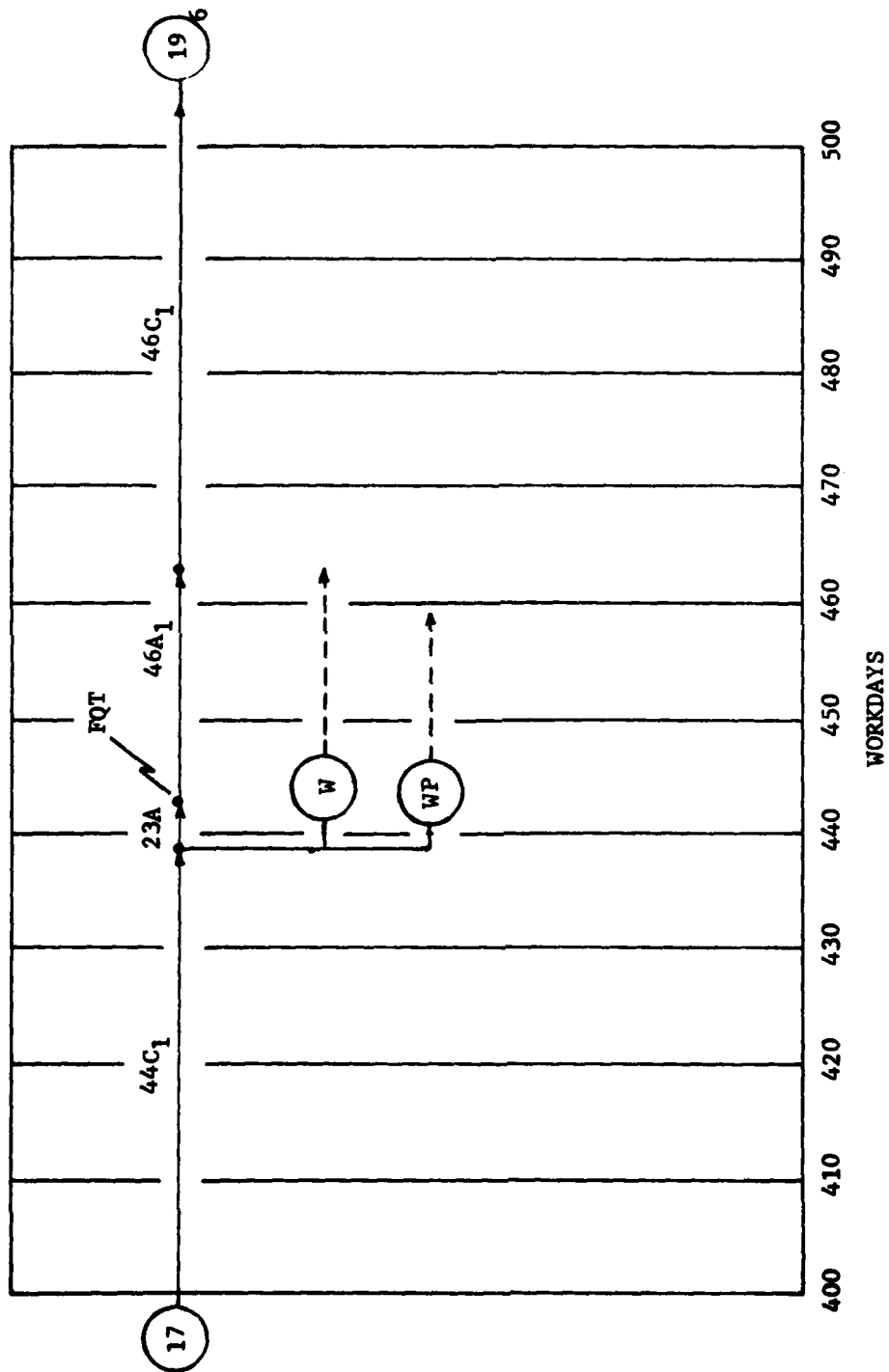


Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 5

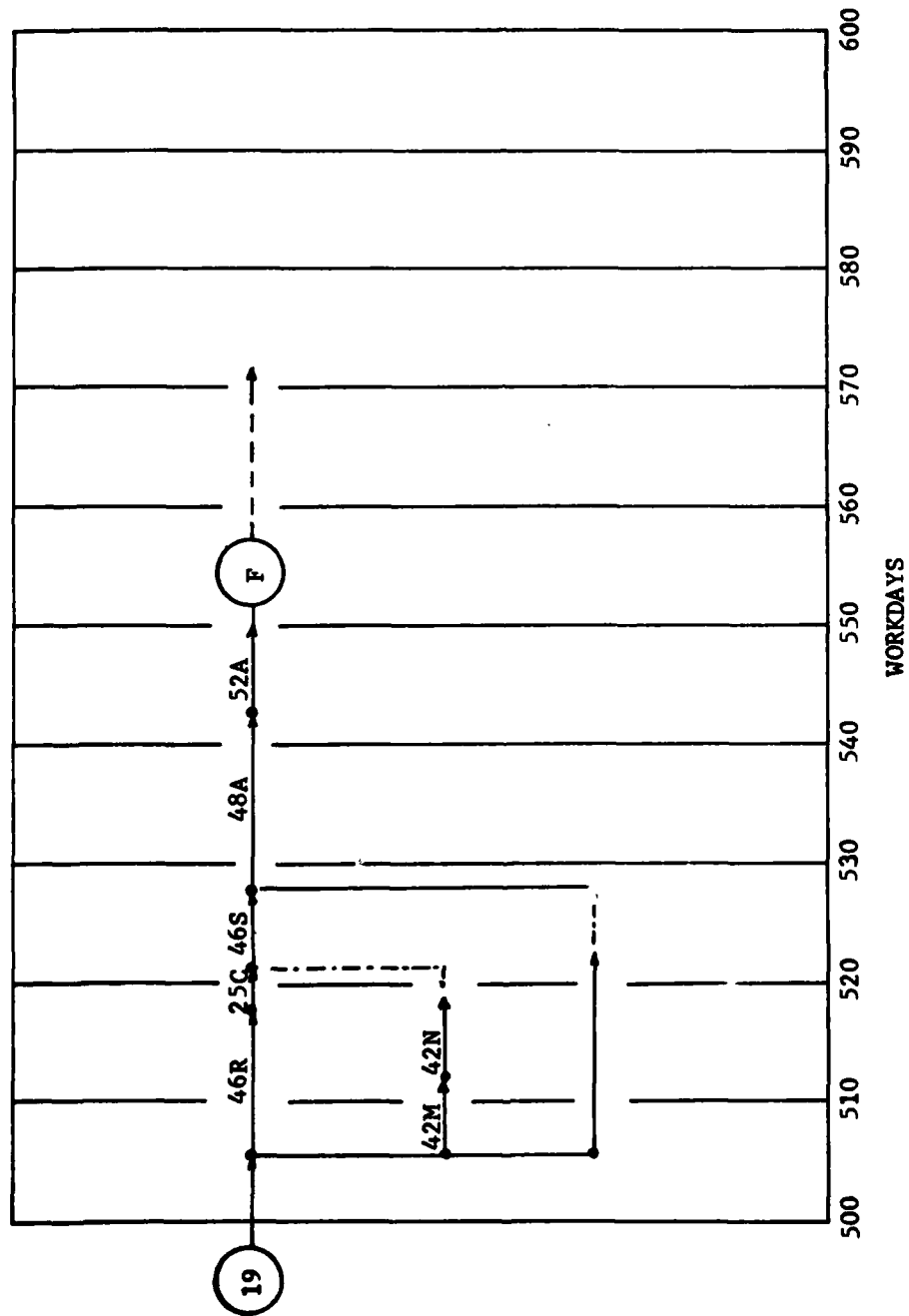


Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 6

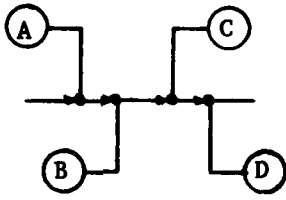


## DIAGRAM NOTATION

### 1. General:

Abbissa shows elapsed time (workdays) from start of contract.  
Critical path is shown on the uppermost path.  
The diagram was drawn to show three DIGs and one TIG.

### 2. Drawing Conventions:

Dot	•	is the start of a box.
Arrowhead	→	is the end of a box.
Solid Line	—	is the duration of a box.
Dashed Lines	- - - - -	show paths to activities not depicted on this abridged diagram
Dashed Dotted Lines	- . - . - . - . -	indicate slack time in the dependency sequences
Connectors		<p>Incoming connectors (A&amp;B) extend to the left</p> <p>Outgoing connectors (C&amp;D) extend to the right</p>

### 3. Labeling Conventions:

Box numbers are shown on or near the solid line; subscripts denote group number.

Lettered connectors correspond with those shown on the LoSim Process Flow Diagram, Figure A-2. Interior subscripts denote group number; exterior numerals denote the destination page.

Numbered connectors apply just to this drawing. Exterior numerals denote the destination page.

Figure 11. Program Acquisition: Activity Sequence and Timing Diagram  
Sheet 7

## SECTION 6

### PROCESS SIMULATION

Except where otherwise stated, this section treats the development and characteristics of the initial Simulator version, Model 0. We intend Model 0 to be a relatively simple computer program that proves feasibility and that provides a sound basis for extension. We also intend Model 0 to produce some useful results; i.e., those indicated in Section 6.6.1, Model 0 Output Reports.

The principal Model 0 reports will be profiles of the manpower needed to perform the network's activities, and a schedule of the major activity groups (termed Subnetworks). These reports will reflect logical precedence constraints but not manpower constraints because the latter, although important, appear difficult to model realistically.

The main improvement planned for Model 1 will be inclusion of one or more algorithms to allocate scarce manpower, and to effect delay whenever too little of the right manpower type is available. Model 1 will also produce dollar cost profiles and more elaborate schedules. Models 2 and 3 will each have additional important capabilities. See Section 6.6.2, Future Model Output Reports.

#### 6.1 DEVELOPMENT FACILITY AND SIMULATION PROGRAMMING LANGUAGE SELECTION

We approached the problem of selecting a development facility and a simulation programming language for initial Simulator development as follows. First, selection criteria appropriate to project needs were established. Then, promising candidates about which information was readily available were evaluated using the criteria, and choices were made. We first selected the development facility, because we deemed it more important than any specific simulation programming language, provided the facility supported an adequate language. This choice narrowed considerably the number of candidate languages.

##### 6.1.1 Facility Selection

Six factors were deemed essential to the initial Simulator development:

- a. affordability within project budgetary constraints;
- b. capacity adequate for efficient compilation and execution;
- c. system availability to Simulator development personnel;
- d. turnaround time for Simulator development runs;
- e. quality and availability of help to solve apparent system problems;

f. availability of an adequate simulation programming language.

Because our budget had no funds for commercial computer rental or purchase, we ruled out the use of commercial time-sharing services or block computer time rental. Nor could we purchase a computer. Criterion a. thus limited us to shared use of an Air Force or MITRE computer.

Previous experience has clearly shown that successful discrete event simulation requires substantial main memory and central processor capacity, especially as the complexity of the simulation program grows and the frequency of simulation runs increases. For this reason (criterion b.) we restricted our search to a large-scale computer, ruling out mini-computers, including one otherwise attractive candidate, a MITRE PDP 11/45.

Criterion c. meant that project programming personnel must have the right to use the computer at reasonable times, and that during most of these times they should be able (directly or remotely) to access the computer to enter data, execute programs, and obtain output. As a practical matter, criteria a., b., and c. jointly limited us to three computer systems: the Air Force Geophysics Lab CDC computer, the RADC Honeywell 6180, and the MITRE Bedford IBM 370. Our choice among these candidate computers was based primarily on criteria c., d, and e, since a satisfactory simulation programming language was available, or could readily be provided, for each candidate.

We judged our ability to use the IBM 370 far superior to our ability to use the RADC Honeywell 6180, mainly because we had very limited access to suitable terminals capable of flexible input and the high-volume output that simulation program development sometimes demands. In contrast, we could use several conveniently located CRT terminals directly connected to the IBM 370, and could easily walk to its dispatcher's desk to pick up printouts. Practical use of the Geophysics Lab computer would have entailed frequent, time-consuming trips to Hanscom AFB.

The MITRE Bedford Computer Center also provides expert help in the use of the computer and its system software. Because of its proximity, we decided that this help was superior, for our project personnel, to that available elsewhere. For these reasons we selected the MITRE computer.

#### 6.1.2 Simulation Programming Language Selection

Given the choice of installation, our practical choice of languages comprised PL/1, PASCAL, FORTRAN, GPSS, GASP IV, and SIMSCRIPT II.5. We ruled out PL/1 in part because it has few compilers for non-IBM computers. The Simulator should ultimately be transferrable among several different computer families; this would be difficult unless the Simulator were coded in a language supported for several machines.

PASCAL, despite some attractive features, lacks a well-tested compiler at the MITRE computer facility. This eliminated it from serious consideration, because our budget and schedule precluded our risking serious support software problems.

In practice FORTRAN is as nearly universal as any programming language in use today. However, we ruled it out, at least for early Simulator development, because (like PL/1 and PASCAL) it lacks a timing routine and other specialized features helpful to simulation. We believed (and still believe) that a higher-level language could reduce Simulator development effort.

In contrast, GPSS is a high-level language designed expressly for discrete event simulation. However, it seems unable to support easily the kind of table-driven program design we envisioned. In addition, we have previous experience indicating excessive GPSS storage and execution time requirements as a simulation program grows. GPSS is also limited mainly to IBM computers, which would impair transfer of a GPSS Simulator. For these reasons we rejected it.

GASP IV is really a collection of FORTRAN subprograms (e.g., a central timing routine) designed to simplify development of simulation computer programs written in FORTRAN. Because these routines appear quite helpful, and because GASP IV programs are as easily transferred as FORTRAN programs (e.g., only a FORTRAN compiler is needed), we considered GASP quite seriously. We still deem it a good second choice. However, we selected SIMSCRIPT II.5 over GASP IV because of the former's more easily readable vocabulary (helpful for self-documentation) and better diagnostics. Also, former SIMSCRIPT II.5 users at MITRE recommended it highly and could be consulted for help.

Our experience with SIMSCRIPT II.5 to date has generally been favorable. We plan to continue using SIMSCRIPT II.5 unless severe problems emerge. If they do we may switch to another simulation programming language.

## 6.2 DESIGN OVERVIEW

Simulator Model 0 will consist of three basic parts: (1) the Data Input Processor; (2) the Simulation Conduct Processor; and (3) the Output Report Generator. The three processors will execute sequentially. The first will operate on input data sets defined in Appendix B to produce part of a data base described in Section 6.3. The second (the Simulation Conduct Processor) will be driven by these data. As a result, the Simulation Conduct Processor will develop simulation statistics which it will store in this data base. The Output Report Generator will then edit and print these statistics. The designs of the Data Input Processor and the Output Report Generator are straightforward. The design of the Simulation Conduct Processor is rather unusual (for a simulation program), and thus worth brief discussion here.

The Simulation Conduct Processor will be table-driven. Thus, the complete Model will be defined by a set of table data such as those given in Appendix B. Each of the two-hundred plus boxes included in the LoSim Process Flow Diagram of the Full-Scale Development Phase, discussed in previous sections of this report, is described by an entry in Table B-1 and another entry in Table B-2, Table B-3, or Table B-4. After the Data Input Processor has reformatted these tables, the Simulation Conduct Processor will read the data for the first box, take actions which depend on the box type (e.g.,

Activity Box, Decision Box) using the assigned parameter values (e.g., activity duration or decision probability), and save any data needed to describe the results. It will then proceed to each of that box's immediate successor boxes; these will be processed in appropriate sequence until all boxes involved in that pass through the network have been accessed. Since the Simulation Conduct Processor's path through the network will be determined by Monte Carlo selection of alternative Decision Box exits, a different sequence of box activation and results will be likely on each path. Thus, the program must repeat many times (per another input parameter) to obtain statistically significant results.

This design concept was selected because of several desirable properties.

- a. The Simulator's description of the process is easily changed; therefore, the nuances of each acquisition program, new facts, revised assumptions, alternate policies, etc., can be modeled readily by appropriate changes in the Simulator's inputs.
- b. The computer program is small and straightforward, because only a few general routines are needed to interpret the tables.
- c. The program so written can directly accommodate Process Flow Diagrams and parameter values developed for other acquisition life cycle phases, or even for different processes. For example, it could readily simulate:
  - (1) the other aspects (e.g., the hardware-related and personnel-related activities) of Major System acquisition programs;
  - (2) programs managed per AFR 300-series regulations;
  - (3) acquisitions conducted per Army, Navy or other agency regulations;
  - (4) other processes that principally involve groups of people working toward common goals.

As a result, in contrast to many simulation programs, which have a block of code for each different section of process flow, the Simulator promises to be relatively simple to build, easy to adapt or extend, and inexpensive to maintain.

### 6.3 DATA BASE

A central data base referenced by the three processors is essential to this design concept. The part of the data base that defines the process network, the process parameters, and simulation options, will be created by the Data Input Processor from inputs defined in Appendix B and illustrated in Tables B-1 through B-4. The Data Input Processor's primary function is to transform this input into a structure that the Simulation Conduct Processor (which will execute many times per simulation run) will process efficiently. The structured input data will then guide the Simulation Conduct Processor

through the network, developing and storing the data needed by the Output Report Generator.

The structure and content of the data base so far developed for Simulator Model 0 is shown in Figure 12, Data Base Structure. Extracts from the compiled version are included within Appendix E, Simulation Program Listing Extracts. The figure shows a Main Entry for each box of the Model. This contains information of general utility that occupies a fixed amount of storage. Each Main Entry also contains pointers to as many as five lists (called Sets in SIMSCRIPT II.5) which contain the remaining data for the box, as explained below.

#### 6.3.1 List Structure

The data base structure relies on the list-handling capabilities of SIMSCRIPT II.5. As described below, list structures help to deal with the complex data that characterize the Process Model design. In Model 0 the following lists may belong to each box's Main Entry:

a. A Pred lists all boxes that are immediate predecessors of the owning box. It also defines conditions for starting that box, for iterating that box, or for executing that box for different Integration Groups. A Pred is a list because a box may have a variable number of predecessors.

b. A Yes.Succ lists all boxes that immediately follow a Decision Box's "Yes" exit. A Yes.Succ also lists the immediate successors of an Activity Box or a Special Event Box.

c. A No.Succ lists all boxes that immediately follow a Decision Box's "No" exit. A No.Succ for a Special Event may point to other boxes which are the objects of parameter changes or reset actions. The No.Succ pointers of Activity Boxes are empty.

d. An OTD (Occurrence and Timing Data) lists certain data (e.g., predecessor completion data, start and finish times) that may vary with DIG or TIG. There is one entry in the list for each DIG or TIG. For boxes not involved in DIG- or TIG-repetition, there is one OTD entry.

e. An ATM (Activity Timing and Manpower) lists the estimated manpower levels (by manpower type) and durations necessary to complete each activity. There is a separate ATM Entry for each DIG or TIG, if any. Otherwise there is a single ATM entry for the activity as a whole. Only Activity Boxes have ATMs. The ATM pointers of Decision Boxes and Special Event Boxes are empty.

#### 6.3.2 Data Packing

Figure 12 shows that SIMSCRIPT II.5 permits packing several data fields into the same computer word; e.g., word two of the Main Entry contains seven different data fields. This significantly reduces the storage space needed to contain the data base.

### 6.3.3 Data Base Size

The data base thus far defined in detail for Simulator Model 0 at the LoSim level comprises about 11,000 computer words, as shown in Figure 13, Data Base Size. An average of 57 words of storage is needed per box.

The data elements defined in Figure 13 will store model definition information and some interim run results, plus some overhead (pointers and counters). The latter are needed to preserve the relationships among the separately stored segments of data. By counting the overhead words, the overhead was found to account for about 20% of the total.

In addition, the system will require storage for system and local data plus storage for data extracted for statistical analysis. These storage needs have not yet been defined in detail. Thus, they cannot yet be quantitatively analyzed, but are believed reasonable.

### 6.4 DATA INPUT PROCESSOR

The Data Input Processor will read in the designated Model definition data (per Appendix B), and (in later Simulator versions) data that describe the system environment and output options. It will check this input for format, consistency, and completeness. If the data do not meet all acceptance criteria, the program will output appropriate diagnostic messages; otherwise the data will be restructured and loaded into the data base.

The Data Input Processor will then initialize the Simulator to begin the specified number of passes through the network, and will then transfer control to the Simulation Conduct Processor. A compiler listing that includes the Data Input Processor code written to date is included within Appendix E.

### 6.5 SIMULATION CONDUCT PROCESSOR

This program will traverse the network from the initial element (e.g., Box 2A) through the final elements (e.g., Boxes 82Y and 54V). It will simulate each activity and decision (and will perform every Special Event), along paths selected randomly per decision probabilities. While following the network, it will develop statistics on decision outcomes, delays, demand for resources, resource utilization, iteration, etc.

The Simulation Conduct Processor will repeat its network traversal as many times as specified in an input parameter termed N.Repetitions. During each traversal it may follow a somewhat different path, because different decision outcomes will be selected randomly. The Event Notice technique, provided by the SIMSCRIPT II.5 language, will control the Simulation Conduct Processor as described below.

#### 6.5.1 Event Notice Concept

a. A SIMSCRIPT II.5 Event Notice initiates a defined set of activities at a scheduled simulated time, specified in each Event Notice. The SIMSCRIPT II.5 Schedule verb creates each Event Notice and files it in an appropriate list.

b. The designer can define any number of different Event Types. For each Event Type there is a defined set of parameters, and an associated computer routine (termed an Event Routine) which accomplishes the functions specified for that Event Type. These functions may include scheduling additional Events.

c. SIMSCRIPT II.5 maintains lists (one for each Event Type) which hold Event Notices waiting to be processed. Whenever an Event Notice has been processed it is normally destroyed, and control returns to the SIMSCRIPT II.5 timing routine which searches all the Event Lists to find the Event Notice with the now earliest scheduled time. Special priority logic resolves ties. The timing routine then actuates the earliest Event and sets Simulation Time to that Event's time.

d. Every SIMSCRIPT II.5 program includes a Main Program. Among other things this must schedule at least the first Event and start simulation; i.e., transfer control to the timing routine to select the first Event Notice to be processed. When the timing routine exhausts all Event Notices, it returns to the Main Program.

#### 6.5.2 Event Notice Use

Simulator Model 0 has been designed to use two Event Types which are defined in Appendix F, Event Notice Specification. One of these (Box.Proc) performs the processing associated with the particular box designated in the Event Notice. This one Event Type will be used to process all box types used in the Model.

The other Event Type (Flow) will be scheduled each time Box.Proc completes its processing. Flow will perform any needed termination processing for the designated box, and then update and examine the input status for each of the boxes that immediately succeed the designated box. For each of the successor boxes on which all input conditions have been satisfied, a Box.Proc Event Notice will be scheduled.

#### 6.5.3 Simulation Conduct Operation

The Simulation Conduct Processor will begin after the Data Input Processor has loaded and reformatted the input data, and has created and scheduled an Event Notice for the first box of the Model (e.g., Box 2A). Then the SIMSCRIPT II.5 timing routine will be called. When it searches its Event Notice Lists it will find just the one Event Notice (a Box.Proc for the initial box) inserted by the Data Input Processor. The timing routine will then set Simulation Time to the Event time (which is zero) and call the Box.Proc Event Routine. This will begin a sequence of Event Notice processes



such as that diagrammed in Figure 14, Event Notice Processing Sequence. This figure shows how the Event Notices are linked together to accomplish the actions described in the simulation tables; it provides an abridged view of the process up to PDR #1.

As shown in the figure the first Box.Proc (for Box 2A) will accomplish the functions associated with Box 2A (an Activity Box), and then schedule a Flow Event to occur on day 10 (because (per Table B-2) the activity duration of Box 2A is 10 days). The Flow Event Routine for Box 2A will terminate the Box 2A activity and then (per Table B-1) schedule two new Box.Proc Events (for Boxes 4A & 4S) for immediate action. The Box 4A Box.Proc Event can only update status; it needs another input (from Box 4S) before its activity can begin. When the Box 4S Event does begin it will schedule a Box 4S Flow Event for 15 days later.

The SIMSCRIPT II.5 Controller will now find that the Box 4S Flow Event is the earliest so it will add 15 days to Simulation Time and then invoke the Event. The Flow Event program will now create immediate Box.Proc Events for the three successors of Box 4S: Boxes 4A, 60A, and 62A. Box.Proc processing for both boxes 60A & 62A will begin immediately; for each a Flow Event will be scheduled. A Box.Proc Event for Box 4A will also begin immediately because both of this box's inputs (Boxes 2A and 4S) will be satisfied; it will schedule a concluding Flow Event for Box 4A 10 days hence. The Box 4A Flow Event Routine will then schedule immediate Box.Proc Events for its successor boxes (4C, 6A, 53A, and 53C).

This interlaced Box.Proc and Flow Event processing will continue until there are no more Event Notices to process. Several cases not described above deserve mention. Whenever a Decision Box Box.Proc is processed (as shown for Box 6E) the exit will be selected in zero Simulation Time. The diagram shows that the "No" exit would be selected on first pass (day 72) so the diagram shows the iterative return to Box 6A. The next time 6E would be reached (day 84) the "Yes" exit to Box 6G would be shown.

The other feature involves the handling of specified internal delays; e.g., on Box 4E. In this case, the Flow Event for Box 4C (day 47) will detect the Box 4E Wait and schedule the Box 4E Box.Proc to begin after the (5 day) internal wait period.

#### 6.5.4 Simulation Results Data Collection

During simulation conduct, timing, resource utilization, and decision outcome data will be collected within the data base. A list of all such data planned for Model 0 is provided in Appendix G, Demonstration Model (M0) Statistics Development. In general, data will be collected for individual boxes, for the whole network, and for user-defined network subdivisions (Subnetworks). The data will be collected primarily to support the needs of the Output Report Generator program. Some of the data however, are planned to aid test and verification.

#### 6.5.5 Multi-pass Control

Because all Decision Box processing and most activity duration computation will include probabilistic elements, each pass through the network will yield somewhat different data. Therefore, as described in Appendix G, each simulation run will include many passes through the network to obtain results with statistical significance. The multi-pass control will provide the means for conducting the input-specified number of passes, extracting and aggregating the data from these, and reinitializing the per-pass data base between each pair of passes.

#### 6.6 OUTPUT REPORT GENERATION

When the Simulation Conduct Processor has completed its tasks, many passes through the network will have been completed and raw data reflecting the simulation results will have been accumulated. The Output Report Generator will reduce this data into defined statistical results, and will prepare reports which encapsulate them, based on report selection input parameters.

These output reports will provide information about system entities at three levels, as follows.

- a. The lowest level entities are individual function boxes.
- b. The highest level entity is the total network.
- c. The intermediate level entities are subdivisions of the total network, hereafter termed Subnetworks. Each Subnetwork consists of a group of function boxes which the user may designate to identify some larger (aggregate) function. For example, all the boxes shown on Sheet 3 of the LoSim Level Process Flow Diagram (Figure A-2) could be formed into a Subnetwork which included all the detailed design activities. The Model 0 user will be able to define up to 15 Subnetworks, by marking each box involved with the number of a single Subnetwork to which it belongs.

Whenever applicable, separate data values will be accumulated for those components of the total which result from process iteration. Also, for processes repeated for different Integration Groups, separate values will be retained for each Group. For most statistical data, the mean value, the standard deviation, the number of instances, the minimum, and the maximum value encountered will be available for output.

#### 6.6.1 Model 0 Output Reports

The Model 0 output reports are defined in Appendix G. In general, they will provide the information listed below for each of the following types of system entity:

##### a. Activity Boxes

- (1) the time expended in waiting for all predecessors' input conditions to be satisfied;
- (2) the total time used for internal waiting (i.e. starting delays inherent in the activity);
- (3) the total time used performing the activity;
- (4) the earliest start time and latest finish time;
- (5) the count of the number of iterations; and
- (6) the total of each category of manpower used.

##### b. Decision Boxes

- (1) the predecessor wait and internal wait times as well as the iteration count, as described for item a. above;
- (2) the earliest and latest occurrence times; and
- (3) a record of the exit selections made.

##### c. Subnetworks

- (1) the earliest start and latest completion times (or equivalent occurrence times); and
- (2) the total of each category of manpower used.

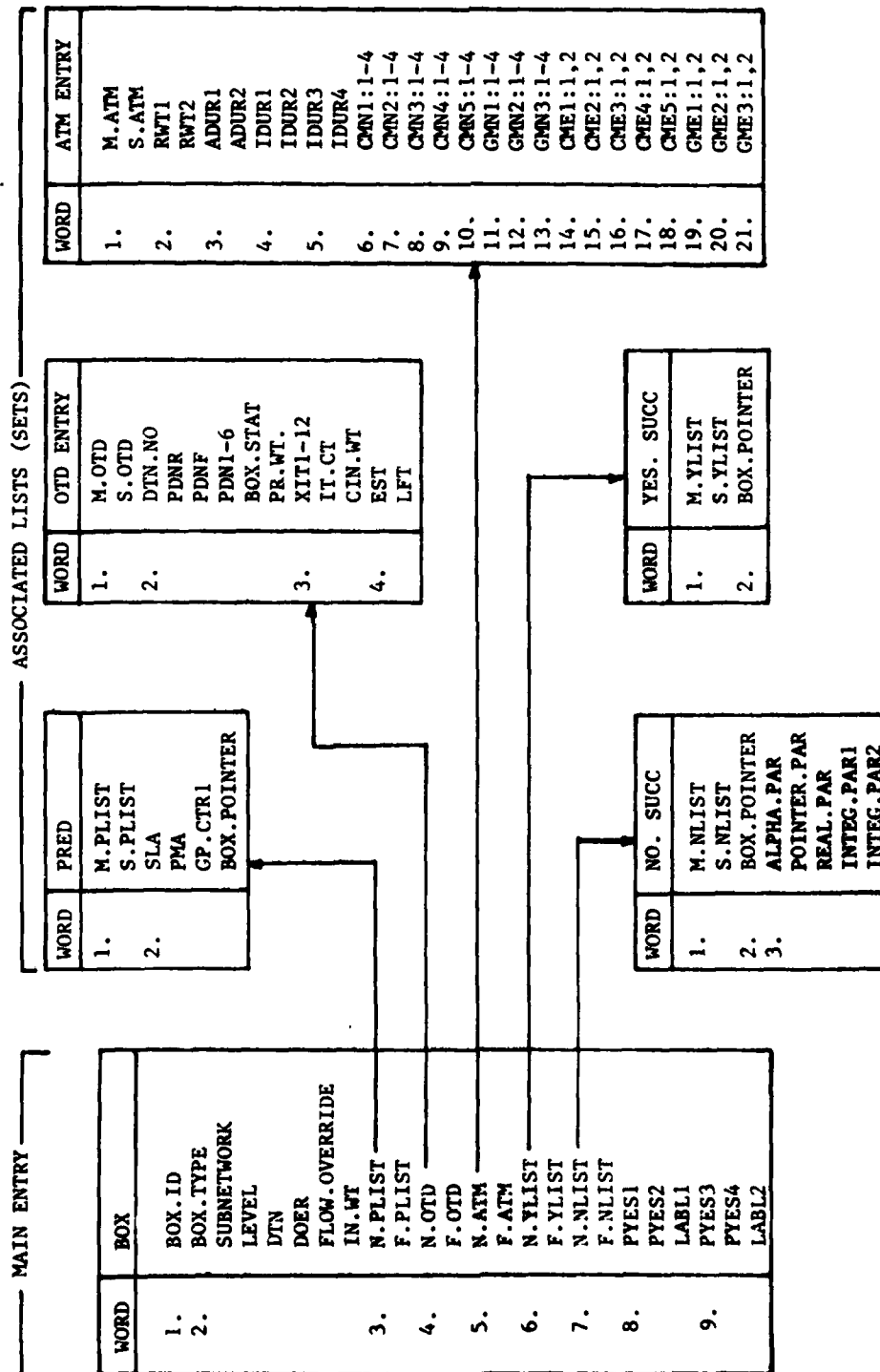
##### d. Full Network

- (1) the total of each category of manpower used;
- (2) a profile of personnel use (i.e., manning level) for each manpower category and total manpower; and
- (3) a project schedule showing milestone times.

#### 6.6.2 Future Model Output Reports

The model simulation technique is capable of producing much more extensive information than that shown for Model 0. Examples of the inherent possibilities are shown in Appendix H, Data Reporting for Later Simulator

Models. The growth in output capability is planned to be evolutionary. The expected user participation during the growth period will help assure that the reports will contain the information needed for the numerous applications seen for the Simulator.



Note: All data names listed in the above boxes are defined in Appendix E

Figure 12. Data Base Structure

The model network at LoSim level includes 191 boxes (not including ECP processing) with the breakdown shown in Figure 4. Data base storage is based on 3 DIGS and 5 TIGS, as follows:

ELEMENT	UNIT COUNT	TOTAL COUNT	UNIT STORAGE (WORDS)	TOTAL WORDS
BOXES	191	191	9	1719
PRED(CESSORS)	286	286	2	572
YES.SUCC(ESSORS)	249	249	2	498
NO.SUCC(ESSORS)	57	57	3	171
OTD(1-Group)	113	113		
OTD(3-Group)	51	153		
OTD(5-Group)	<u>23</u>	<u>115</u>		
TOTAL	191	401	4	1604
ATM(1-Group)	70	70		
ATM(3-Group)	40	120		
ATM(5-Group)	<u>23</u>	<u>115</u>		
TOTAL	133	305	21	6405
TOTAL BOX DATA STORAGE				<u>10,969</u>
AVERAGE STORAGE PER BOX - - - - -				57 WORDS

Figure 13. Data Base Size



## SECTION 7

### ACCOMPLISHMENTS AND STATUS

#### 7.1 ACCOMPLISHMENTS

The work to date on the Software Acquisition Process Model was conducted and documented in three major time segments. During the initial phase, the software acquisition problem was defined, prior art solutions were evaluated, the Process Model concept was described, a number of applications were identified, and Model feasibility was explored. The results of that work indicated that the Model was feasible and desirable.

During the second phase of the project, major emphasis shifted towards development of a diagrammatic representation of the acquisition process which could support the mechanization of the Model. Process Flow Diagrams in three levels of detail were developed to a point where they could be compared and evaluated. It was concluded that the middle level was best for simulation, while the other levels had supplementary value. During this period, the initial Simulator programming language selection decision was made and some exploratory coding was accomplished. Finally, the parameter value estimation technique was selected and the decision was made to limit the initial scope of the effort to the Full-Scale Development Phase of the overall acquisition process.

With the preliminary investigation and decisions completed, the final portion of the work to date was concentrated on development of a satisfactory design concept and initial implementation of the design. The Model definition process was refined and essentially completed. The quantification process was extended to allow estimated parameter values to be refined by calibration. During this final period, a very promising set of Simulator design concepts was evolved and adopted. Much of the very rapid progress described in the technical portions of this document are attributable to this design. As a result, a successful outcome to this project is most likely and the problem solving potential of the Simulator still appears to be viable and realizable.

#### 7.2 CURRENT STATUS

The current status of the principal project tasks is described below. All the status information applies to the work on Simulator Model 0 and the Full-Scale Development Phase of the Major System acquisition process, which has been the focus of the FY 79 effort. While quantitative terms have been used to express the degree of completion on some tasks, these are necessarily approximate. Also, some work shown to be complete might require revision later, because the close interrelationship between the various tasks could cause new work to perturb previously completed work.



#### **7.2.1 Process Model Definition**

- a. Overview Diagrams: completed
- b. Simulation Level (HiSim and LoSim) Diagrams: completed
- c. Expanded View Diagram: 20% completed
- d. Amplification Notes: 20% completed
- e. Network Linkage Table: completed.

#### **7.2.2 Process Model Quantification**

- a. Parameter Selection: completed
- b. Parameter Value Estimation:
  - (1) Decision Probabilities: completed
  - (2) Activity Timing: 50% completed
  - (3) Manning Levels: 25% completed.
- c. Parameter Calibration:
  - (1) Concept: complete
  - (2) Timing: main threads (critical path only) completed
  - (3) Manning: 10% completed.
- d. Parameter Tables:
  - (1) Structure and Definition: LoSim level complete
  - (2) Data Content: all current estimates are entered.

#### **7.2.3 Process Model Simulation**

- a. Data Base Definition: completed for all box data
- b. Data Base Coding: completed for all box data
- c. Model Control Concepts and Structure: completed
- d. Model 0 Outputs: defined
- e. Future Model Outputs: tentative definition prepared
- f. Event Notice Program Definitions: completed for all box data (except for Special Event Boxes)
- g. Data Input Processor Coding: mostly exploratory; 5% completed.

## SECTION 8

### ANTICIPATED GROWTH AND PLANS

#### 8.1 GROWTH AREAS

As part of the phased implementation approach being used on this project, Model 0 includes a central core of essential capabilities with built-in growth support features. Listed below are additional capabilities which are likely candidates for inclusion in later Models. The growth areas have been grouped into the task areas (i.e., Definition, Quantification, and Simulation) previously used throughout this report. It is to be understood, however, that each new capability will probably have some impact on all three areas.

##### 8.1.1 Model Definition

a. The LoSim Diagram (Figure A-2) needs review by others and refinement. It should be simplified by combining some lesser functions into larger ones. It should provide for more flexible Integration Grouping; i.e., some difficult tasks should be permitted to begin in Group 1 and to conclude with a later group.

b. The FSD Model should be made to represent interdependencies among CPCIs and also with any interfacing equipment Configuration Items (CIs) which are being concurrently developed or procured.

c. Interfacing Models should be created for the other Major System Acquisition Life Cycle phases, covering the embedded software acquisition process; i.e., the Validation, Conceptual, Production, and Deployment phases, probably in that order.

d. The Modeled process should eventually cover the procurement of entire Electronic Systems, including the acquisition and development of special hardware CIs.

e. The representation of the ECP process needs to be improved so that its impact on and relationships with the other activities and discussions becomes a reasonable reflection of actual practices.

##### 8.1.2 Quantification

a. All currently used numerical values, which reflect "typical" usage should be replaced by generic functional relationships, which can therefore be applied to a large variety of contractual and developmental situations; see Section 5.5.

b. The generic relationships should be developed in conjunction with these developed on the SARE project; see Section 5.5.4.

c. The time, resource, and probability quantities associated with an activity or decision should be made to reflect the quality of prerequisite prior activities and decisions. While the quality of an intermediate product (i.e., an output that is produced by an Activity Box) is not directly assessed, it can be inferred in some cases from the quantity and quality of the effort expended. I.e., if person (or group) with a given capability requires thirty days to properly accomplish a task, but actually uses fifteen days, the Model could assume that the output product is poor. Therefore, any decisions based on the product should be biased toward non-acceptance and any subsequent actions which use the product should be biased towards longer duration or poorer quality. Similarly, if an acceptance decision resulted in iteration of the activity (i.e., rework), the output product quality would normally be improved. This quality dependency, while complex and difficult to quantify, is necessary if the Model is ever to achieve a life-like representation of the acquisition process. The draft Data Item Description for SARE reporting includes parameters that define quantitative measures of quality (and other factors that affect software time and resource consumption). We expect that these, or variations of them, can be used in the Model.

d. Resource utilization parameters in addition to manpower should be included. In particular, facilities used for program development and for test must each be quantified to reflect its maximum utilization rate as well as the rate of use by each pertinent Activity Box. The Model must include capability to represent separate quantities of as many kinds of resources as may be used.

e. The number of manpower types assigned to each task may need to be changed to reflect subdivision beyond those currently used; see Section 4.1.7.

f. No dollar cost data are developed in Model 0. Dollar cost data reflecting manpower use should soon be added, and that reflecting other resources should follow shortly. The dollar cost data will also need to reflect the changing value of money during the acquisition period.

### 8.1.3 Simulation

When implemented, all changes described above (e.g., those regarding quality) will impact the Simulator. A few deserve special attention as provided below.

a. The initial Model will maintain records of manpower usage, but will not limit the quantities available. Since the resources available usually constrain real projects, later versions of the Model will need to retain quantitative data on the availability limits for each kind of resource. This situation requires that the Model eventually reflect the following:

- (1) Manpower availability varies with time as a result of factors such as planned build-up and reduction, employee transfer or resignation, and sickness.

- (2) Other resource availability also varies with time due to equipment acquisition time, equipment maintenance and unscheduled down time, contention with other in-house projects, etc.
- (3) On any real project, the allocation of scarce resources among competing activities is resolved by real-time management action. In a simulation model, the allocation is usually resolved by a management strategy function. Since many different management strategies may be used including complex "look ahead" functions, the development of this area is planned to proceed in an evolutionary manner. Therefore, early versions will use simple, primitive strategies (e.g., wait until all needed resources are available); later ones will allocate and move resources on the basis of impending urgency, including the effects of transiency.

b. At various points during a project, a completed event can trigger a multiple burst set of like activities. E.g., completion of a CDR leads immediately to simultaneous coding by many programmers on many program modules. The initial Model treats these situations as randomly variable phenomena, with each burst having one beginning and one end. On real developments, the many unit tasks proceed with different rates and with differing degrees of success. The real world can therefore show integration problems (e.g., a critical module is not ready) which are not reflected by the initial Model. While it is not considered practical to model burst activities at a level which reflects each worker's accomplishments, it will eventually be necessary to find a modeling level which does represent multiple bursts realistically.

c. The modeling of the development of multiple interdependent CPCIs can be handled by the Simulator Model 0 design. Each CPI can be separately treated, with each independently following its normal process flow diagram, with its own parameter values, function box numbers, and idiosyncratic adjustments. To do this data which define all the CPCIs will be included in the function box definition tables defined in Appendix B. All precedence dependencies among the CPCIs will then be reflected by inclusion in the Network Linkage Table (Table B-1); this treatment is therefore the same as for intra-CPCI dependencies. However, later Simulator versions may incorporate a way to treat multiple CPCIs more compactly.

d. The simulation of ECP behavior does create special simulation problems, such as:

- (1) ECP occurrences on real contracts are generally in response to problem situations which cannot be forecast on an individual basis. Certain periods during the development, however, are more prone to produce ECPs. The planned Simulator design provides for spontaneous generation of ECPs at a controllable rate which is randomized when applied.
- (2) The generation of an ECP consumes intrinsic effort which is easily modeled. It can also profoundly effect the time and costs of

development activities not yet reached or may require rework on already completed activities.

These effects can be simulated in the Model by the following means. Special Event Boxes in the Model's ECP processor will alter the time and effort parameters of the function boxes which would be affected by the ECP, and that may still be reached. The ECP processor itself will include function boxes to emulate any rework needed as a result of each ECP. The time and effort impacts will have to be treated as widely varying random variables which reflect empirical data gleaned from other like contracts.

#### 8.1.4 Output Reports

The Simulator is planned as a general tool for the support of acquisition programs that include software development. As such it can be useful for project planning, proposal evaluation, contract monitoring, acquisition strategy research, and development strategy research. The users' need for information will reflect the specific application intended. While the initial Simulator version will produce limited reports, future versions will provide many more. For this reason, the Simulator will be designed to provide different types of output reports, and will allow operator inputs to specify and delimit specially desired data.

### 8.2 PLANNED FY 80 ACCOMPLISHMENTS

The completion of the Simulation Model remains a multi-year effort during which a sequence of increasingly capable Models will be developed and put into use. Assuming funding at the level of 2 MTS and appropriate computer time, we plan to do the following in FY 80. The project's overall plan, covering past, current, and future work, is discussed in Section 8.3.

The principal Simulator developments planned for FY 80 are completion of a Demonstration Model (M0) and establishment of a firm foundation for the Prototype Model (M1). In addition, the Model Definition work will be extended in depth.

#### a. Demonstration Model (M0)

- (1) Simulator Model 0 will be completely coded, compiled, integrated and checked out.
- (2) Model quantification will be completed; parameter value estimates will be provided for all LoSim function boxes listed in Appendix B tables. Parameter calibration will begin after Simulator Model 0 becomes available, but only as time permits.
- (3) Model 0 will be demonstrated, but no formal test (e.g., FQT) is planned.
- (4) Model definition will be completed by general refinement of the LoSim logic (Figure A-2). Also, the Process Flow Expansion

diagram (Figure A-3) and the Amplification Notes (Table A-2), will be completed.

b. Prototype Model (M1)

- (1) The specific capabilities to be included within Model 1 will be selected and functionally specified.
- (2) Process Flow Diagrams will be updated as necessary to reflect the added capabilities.
- (3) Estimated parameter values will be obtained for any functions modified or added.
- (4) The Simulator design will be updated to include the new capabilities.
- (5) The new capabilities will be implemented (coded, compiled, integrated, etc.) to the extent that available resources permit.

### 8.3 OVERALL PLAN

A modest amount of effort has been devoted to outlining an overall strawman plan for the development, pilot application, and installation of the Process Model, including work in future years. This outline is presented in Table 1, Outline of Strawman Software Acquisition Process Model Development Plan. It needs more work to become a full-fledged plan, and essential concurrence by all concerned organizations to become a viable one. Nevertheless, we hope that the outline will help avert misunderstandings about our goals, our approach, and the scope of our current effort. We also intend it to help guide that effort, and believe that it should lead to more realistic expectations and financial planning.

Our outline draws extensively on ideas contained in Sections 2 and 6 of the 1 April 1979 draft AFSC Software Cost Estimation Working Group (SCEWG) Research Management Plan (SCEWG79), prepared by Capt. J. A. Duquette, ESD/ACCE, based on review of an earlier version by the ESD/SCEWG. However, we have restructured and revised that plan's elements, and have rescheduled the revised results, in preparing our plan outline.

The strawman plan outline has several prime characteristics, all of which we deem essential to the Model's successful development and transition to operational use. These characteristics are:

- a. definition, development, and trial application of several successive, modest, versions of the Model;
- b. incorporating experience with earlier versions into later versions;
- c. strong emphasis on written application guidance and expert support of Model operation and maintenance; and

d. formal review and evaluation of application experience to assess progress, to assure operational effectiveness, and to help guide new version development.

Table 1 lists the major tasks and first-level subtasks of the strawman plan outline, including tasks already accomplished as well as the FY 80 tasks described in Section 8.2. It also incorporates a rudimentary schedule for each. The plan outline has been devised around four potential Software Acquisition Process Model versions. Table 2, Tentative Later Process Model Version Characteristics, presents a number of characteristics of the last three versions, as we currently envision them. In addition, each version will undergo improvement during its pilot application, and the third and fourth versions will change during their routine (production) application.

A principal objective of the plan outline is useful application of the Process Model at the earliest feasible time. Another goal is assuring adequate transfer of experience with early versions into the designs of their successors. A third aim is to define versions of modest difficulty, each of which can be developed with reasonable effort and risk, and each of which can replace its predecessor smoothly.

The versions identified in Table 2 and the schedule in Table 1 are based on planning to date. More thorough description of each subtask may reveal additional opportunities for overlapping tasks, but may also show additional constraints on overlapping. After general agreement is reached on this outline or a revised one, a plan based on it should be developed, as follows. The outline should be expanded, the subtasks should be described, and their mutual information requirements should be elicited. The results should be checked for precedence conflicts. The individual subtasks' resource requirements should be estimated, and an improved plan (including a revised schedule) should be prepared, consistent with available resources. This plan should be completed and agreed on before firm commitments are made about FY 81 work, and updated at roughly six-month intervals thereafter.

Subsequent paragraphs contain preliminary information about selected portions of the plan.

#### 8.3.1 Feasibility Analysis

We completed this task in November 1978. Its chief product is an internal report entitled "ESD Software Acquisition Process Model Concept & Feasibility."

#### 8.3.2 Demonstration Model (M0)

This is the first of the four Model versions. Its development should validate the Simulator design concepts (as described herein), establish basic capability to which desired improvements may be added, and provide valuable simulation experience.

a. M0 Definition

This work is completed. This report outlines Model 0's structure and capabilities.

b. M0 Development

This task is currently in process. This report describes the results to date and our plans for its completion.

8.3.3 Prototype Model (M1)

a. M1 Definition and Preliminary Design

The functions planned for inclusion in M1 are indicated in Table 2. A preliminary design for implementing these functions will lead to a decision during FY 80 as to which functions are appropriate for M1.

b. M1 Development

We plan to start this task in FY 80. The effort will consist of the following subtasks:

(1) Detailed Design

We see this as a conventional computer program design subtask. However, we plan only informal design reviews, since a computer program of this modest magnitude needs no elaborate PDR & CDR.

(2) Coding & Checkout

As usual, this will cover coding, compilation, and unit testing.

(3) Informal Test & Integration

This subtask will cover integration and testing of Simulator Model 1 as a whole. For a program of this modest size and difficulty, developed under close customer observation, we plan no formal testing (i.e., PQT or FQT).

(4) Preparation of Products

Selected output reports, related inputs, code listings, and a set of computer program operating instructions will be prepared and delivered. The computer program source code will also be delivered on magnetic tape. The computer program operating instructions will explain the mechanics of specifying the program's inputs and running it.



#### c. M1 Pilot Application Preparation

Preparation for the initial pilot application entails the following:

- (1) helping ESD to locate existing ESD acquisition programs which might benefit from trial Prototype Process Model application; the effort will include preparation and frequent delivery of a briefing explaining why a PO should consider Prototype Process Model use;
- (2) helping ESD to negotiate a mutually satisfactory arrangement for an initial pilot application with one of these acquisition program POs;
- (3) developing a technical methodology for effective PO use of the Prototype Process Model; and
- (4) together with pilot PO personnel, developing operating instructions for use of the Model.

#### d. M1 Pilot Application

This task covers the application itself, and the required support. Such support includes: (1) helping PO personnel to define Model data and logic changes; (2) Simulator execution, maintenance, and modification during pilot application; (3) interpreting results to PO personnel; and (4) collecting data for pilot application evaluation.

#### 8.3.4 Initial Operational Model (M2)

This is the first of the four proposed Process Model versions (see Table 2) for which we propose routine operational use at ESD. We plan definition, development, pilot application preparation, and pilot application for the Initial Operational Model. These subtasks are analogous to the corresponding Prototype Process Model (M1) subtasks. The chief differences now apparent result from the increased scope and sophistication of the Initial Operational Model, the reduced effort at basic definition needed, plus additional effort that will be required to improve the accuracy of parameter estimates and to incorporate other improvements suggested by Prototype Process Model pilot application experience.

#### 8.3.5 Assessment of Initial Pilot Applications

We plan a several-month effort in early FY 82 to review the Prototype Process Model pilot application experience, and early experience with the Initial Operational Model, to assure that continuation of the effort into routine operation is worthwhile. Assuming a positive outcome, the assessment should also yield recommendations that influence such routine operation.

### 8.3.6 Routine ESD Model Operation (M2 & M3)

To support routine use at ESD of the Initial Operational Model (M2) and later the Extended Model (M3) we plan the following tasks.

#### a. Planning, Training & Support

A permanent organization or organizations, responsible for coordinating SARE data collection, Process Model application, and analysis of related information must be established. We expect to devote modest effort over an extensive time period to plan this organization's establishment, and to support it technically.

#### b. Model Operation at ESD

Support of PO applications is the work envisioned under this subtask.

#### c. Model Maintenance & Modification

Trouble-shooting of the Simulator and its documentation, plus development and installation of more extensive future modifications, are the work planned.

#### d. Transition to Other Users

Assuming successful use of the Model at ESD, its use at other AFSC Product Divisions is expected. The effort planned for this subtask is support for such transition. This support will include installation at other sites (including site adaptation), and help with initial applications at these sites.

### 8.3.7 Extended Model (M3)

This is the fourth and last Model version included in the plan (see Table 2). The effort foreseen to develop and install M3 is analogous to the Initial Operational Model subtasks. The principal difference now foreseen is additional work needed to replace the Initial Operational Model in routine use, which will probably mean an extended period of parallel operation.

### 8.3.8 Model Review and Evaluation

This is planned as a formal evaluation of the entire Model development and application effort by an outside group. Our planned effort comprises support of the evaluation group.

Table 1

## Outline of Strawman Software Acquisition Process Model Development Plan

Task	Title	Schedule (FY Quarters)							
		78	79	80	81	82	83	84	
		1234	1234	1234	1234	1234	1234	1234	
1.	FEASIBILITY ANALYSIS	***	*						
2.	DEMONSTRATION MODEL (M0)								
2.1	M0 Definition		***						
2.2	M0 Development		*	***					
3.	PROTOTYPE MODEL (M1)								
3.1	M1 Definition		**	*					
3.2	M1 Development			***					
3.3	M1 Pilot Application Preparation			*	**				
3.4	M1 Pilot Application				***	**			
4.	INITIAL OPERATIONAL MODEL (M2)								
4.1	M2 Definition			*	**				
4.2	M2 Development				***				
4.3	M2 Pilot Application Transition				**				
4.4	M2 Pilot Application					***	***		
5.	ASSESSMENT OF INITIAL PILOT APPLICATIONS					**			
6.	ROUTINE ESD MODEL OPERATION (M2 & M3)								
6.1	CSDCA Planning, Training & Support					***	***	***	
6.2	Model Operation at ESD					**	***	***	
6.3	Model Maintenance & Modification					**	***	***	
6.4	Transition to Other Users						***	***	
7.	EXTENDED MODEL (M3)								
7.1	M3 Definition					***			
7.2	M3 Development				*	*	**		
7.3	M3 Pilot Application Preparation					**	**		
7.4	M3 Pilot Application						**	***	
7.5	M3 Transition to ESD Operation						*	*	
8.	MODEL REVIEW & EVALUATION							*	

Table 2

## Tentative Later Process Model Version Characteristics

Characteristic	Prototype Model	Initial Operational Model	Extended Model
Breadth (phases covered)	Full-Scale Development.	Full-Scale Development, Validation, & Conceptual.	All phases.
Parametric relationships	No explicit parametric relationships.	Time & manpower joint functions of job size & difficulty.	Time, resource needs & decision probabilities joint functions of SARE Parameters.
Parameter values	Model definers' initial estimates, calibrated vs. typical system. Some MITRE & ESD review.	Initial calibrated estimates revised per government & industry review, some pilot application experience, & some DACS data.	Use of latest data from DACS, SARE, plus experience gained through usage.
Process interdependencies	Few; e.g., ECP frequency changes.	Intermediate; e.g., within local loops.	Full; e.g., an activity product's quality affects resources, time & probabilities of activities & decisions that use the product.
Resource types & allocation	Manpower only. FIFO & priority allocation.	Manpower & facility capacity Allocation is a function of scarcity.	All types. Multiple allocation
System alternative representation	Multiple lower-level entries treated as single entities.	Separate representation of multiple CPCs & their interactions.	Same as initial Operational Model.
Run-time options selectable	A few standard outputs only.	More outputs, most parameter values, no logic.	Extensive output & run controls to meet the defined needs of the user.

## SECTION 9

### CONCLUSIONS AND RECOMMENDATIONS

Effective system planning requires a reasonably accurate means for estimating the cost and schedule for embedded software. This need became manifest about twenty years ago when the first Air Force "L" systems were being acquired. Since then, system technology has greatly improved, enabling the feasibility of ever more complex systems. Software estimating and management, however, remain rough empirical procedures with outputs that lead more to surprises than to sound planning.

While the various analytic estimating techniques currently being used do yield estimates, they have not been able to forecast or account for the wide deviations in results experienced on different but "similar" systems. Because of this situation, work was begun on a software estimating technique that is based on a simulation of the acquisition/development process rather than just on the product being acquired. Though more complex than the analytical methods, this approach appeared to offer the prospect of better results.

Now, 21 months later, the project is closer to realization and the prospect remains bright. The software acquisition process has been modeled, a Simulator design has been formulated and partially implemented, and a plan has been drawn up for bringing the Simulator into operational use. Based on the problems encountered and surmounted, and the results achieved, the Simulator concept has grown in feasibility and still offers the prospect of improved software cost and schedule forecasting.

In addition, during work on the Software Acquisition Process Model, it became evident that the device had inherent capability as a general purpose acquisition management tool. Beyond cost and schedule estimating, it could aid in project planning, contractor proposal evaluation, contractor monitoring, and personnel training, as well as in research into the process of acquiring and developing embedded software.

The results of inaccurate software estimates and dimly illuminated management decisions are manifest in the high cost of acquiring embedded software. Considering the magnitude of annual Air Force expenditures on such software, improvement in the software acquisition process provides considerable potential for cost savings. The Process Model described in this report offers such an opportunity. Since the project cost is small, its cost saving potential large, and the project risk modest, continuation of this work is prudent and recommended.

APPENDIX A

PROCESS FLOW DIAGRAMS  
AND AMPLIFICATION NOTES

This appendix incorporates and explains the detailed diagrams of the software-related activities and decisions typical during the Full-Scale Development Phase of the Major System Acquisition Life Cycle defined in AFR 800-2. As such, it presents the results to date of the Process Definition work, introduced in Section 3 and discussed further in Section 4.

First, Figure A-1, Flow Diagram Notation, explains the flow diagram conventions. Table A-1, Index to Figure A-2 Connectors and Box Numbers, is provided to help locate specific information in the multi-page LoSim flow diagrams. Figure A-2, Software Acquisition Process Model LoSim Activity Flow, depicts in 200+ connected activities and decisions, the software-related functions of the entire Full-Scale Development Phase. Figure A-3, Process Flow Expansion, shows selected portions of this process in detail suitable for training or the development of parameter estimates. Table A-2, Process Flow Diagram Amplification Notes, provides comments on selected activities and decisions depicted in Figures A-2 and A-3. Finally, the abbreviations used in this appendix are listed and defined.

The Process Flow Diagrams contain three basic types of element; Function Boxes, Auxiliary Elements and Lines of Flow, as follows:

## 1. FUNCTION BOXES

### 1.1 Shapes



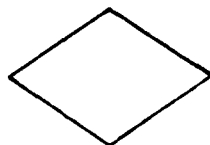
In HiSim and higher-level Process Flow Diagrams, rectangles are used to represent all actions; these diagrams include none of the other Function Box shapes, shown below. In LoSim and Expansion Process Flow Diagrams, however, rectangles (i.e., Rectangular Activity Boxes) are used only to represent mainstream activities (i.e., activities of principal importance).



Trapezoidal Activity Boxes are used to represent support activities in LoSim and Process Flow Expansion Diagrams. Both mainstream and support activities consume time and resources.



In LoSim and Process Flow Expansion Diagrams, a hexagon depicts a Special Event Box. A Special Event provides for special actions such as changing a frequency, duration, or condition, at a designated point in the process logic.



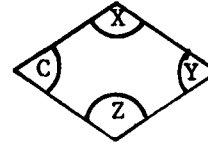
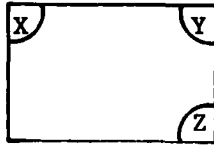
In LoSim and Process Flow Expansion Diagrams, a rhomboid depicts each Decision Box. A Decision Box is any procedure which selects between two mutually exclusive exits. By convention, these include no time or resource expenditures, which are included instead in preceding activities.

### 1.2 Labels

Each Function Box has a label, printed just above the box. In the HiSim diagram each label is a one- or two-digit number. In the LoSim diagram, each HiSim box may be represented as a network of several boxes; thus, each LoSim box label is normally the corresponding HiSim box label suffixed by a letter. Similarly, each Process Flow Expansion box label is normally the corresponding LoSim box label, suffixed by a distinguishing integer.

Figure A-1. Flow Diagram Notation

### 1.3 Features



Each box may contain several field designators, identified by corner positions within the box as shown by letters X, Y, Z and C, as follows:

X - indicates Doer; i.e., the organization responsible for the function:  
A = government (e.g., Air Force), C = Contractor, B = Both.

Y - indicates Integration Group: D = Developmental Integration Group (DIG), T = Test Integration Group (TIG), Blank = the function is not divided into Groups.

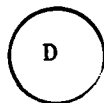
Z - indicates the level at which the work is conducted: 1 = System, 2 = Segment, 3 = CPCI, 4 = CPC (Computer Program Component), 5 = lower level module.

C - is present on any Decision Box used as a counter.

## 2. AUXILIARY ELEMENTS

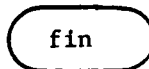
### 2.1 Shapes

Connector



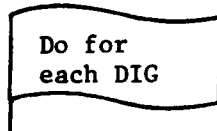
Used to indicate a specific point in the process flow. May be used to show connection between physically separated elements on flow diagrams. (A given label must apply uniquely to only one input point in the process flow).

Terminus



Used to mark a start or end point of a process. When labelled "fin" it marks the end of the specific flow path.

Flag



Used to annotate flow diagrams.

Figure A-1. Flow Diagram Notation (Continued)



### 3. LINES OF FLOW

The lines of flow have arrows to indicate direction, plus three alphabetic designators, as follows:

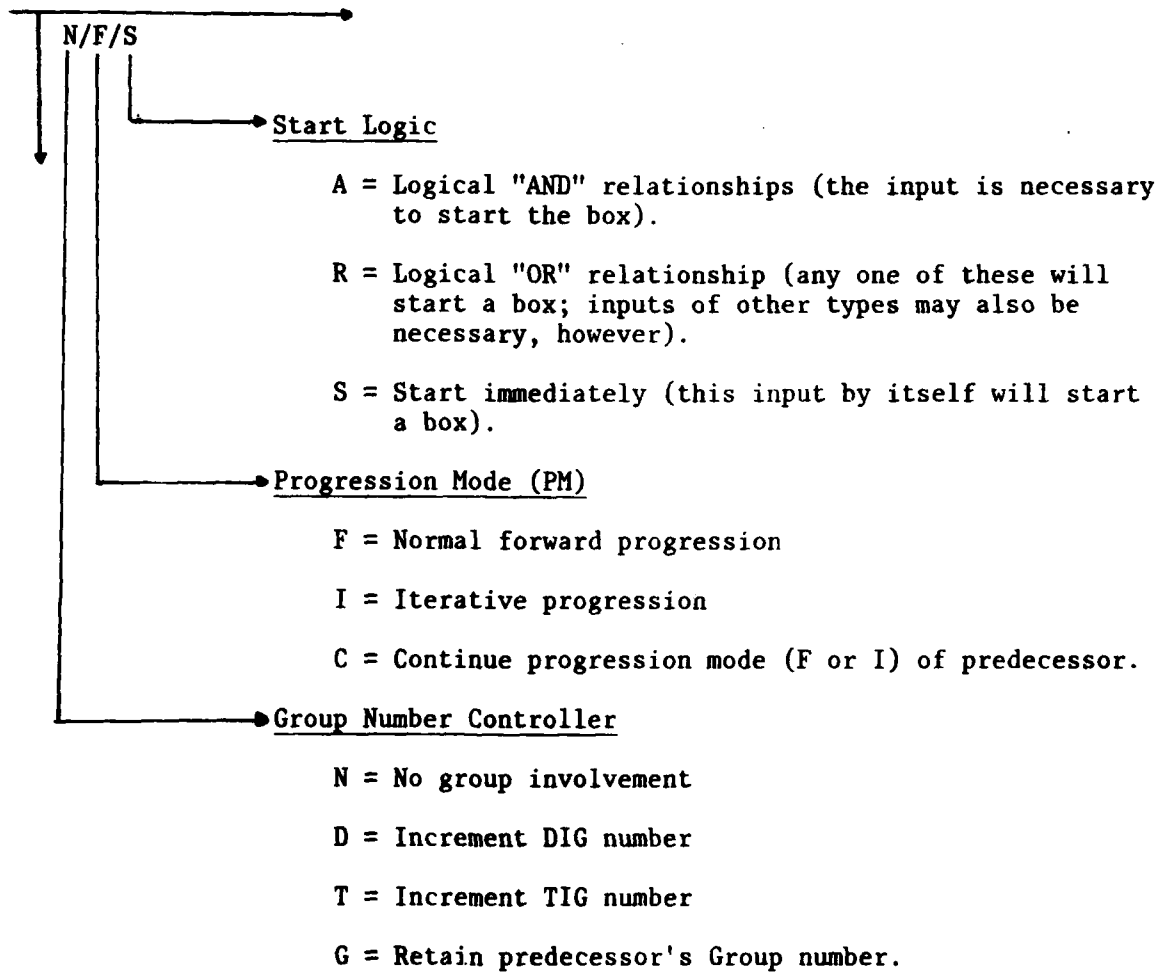


Figure A-1. Flow Diagram Notation (Concluded)

Table A-1

## Index to Figure A-2 Connectors And Box Numbers

CONNECTOR INDEX																			
Connector	A	AA	AC	AE	AQ	B	C	CC	CD	D	DC	DF	DX	E	E1	E2	F	FC	L
Input Sheet	9	9	10	9	10	1	4	4	4	3	3	8	4	12	12	12	7	7	2
Output Sheet	1,8	7	9	8	9		3,4	1	1	2,3	8	3	2	12	12	12	6		1,2

Connector	LC	LE	LP	P	Q	R	RQ	RT	T	U	UP	V	VF	W	WP	X	Y
Input Sheet	2	8	2	5	11	5	11	11	6	8	8	6	6	8	8	11	10
Output Sheet	2	2	8	2	11	1,4	5	6	5,6	6	6		7	5	5	1	1,11

BOX NUMBER INDEX																	
Box No.	2	4	6	8	10	12-	12H	14	16	18	20	22	23	24	25	26	28
Sheet	1	1	2	2	3	3	8	4	4	4	8	4	5	8	6	8	8

Box No.	40	42-	42	44	46	48	50	52-	52A	53	54	60	62-	62C
		M&N												
Sheet	5	5	6	5	6	6	11	7	6	11	11	1	1	4

Box No.	66	70	72	74	80	82	82	96
						A-G	J-Y	
Sheet	1	8	8	8	9	9	10	12

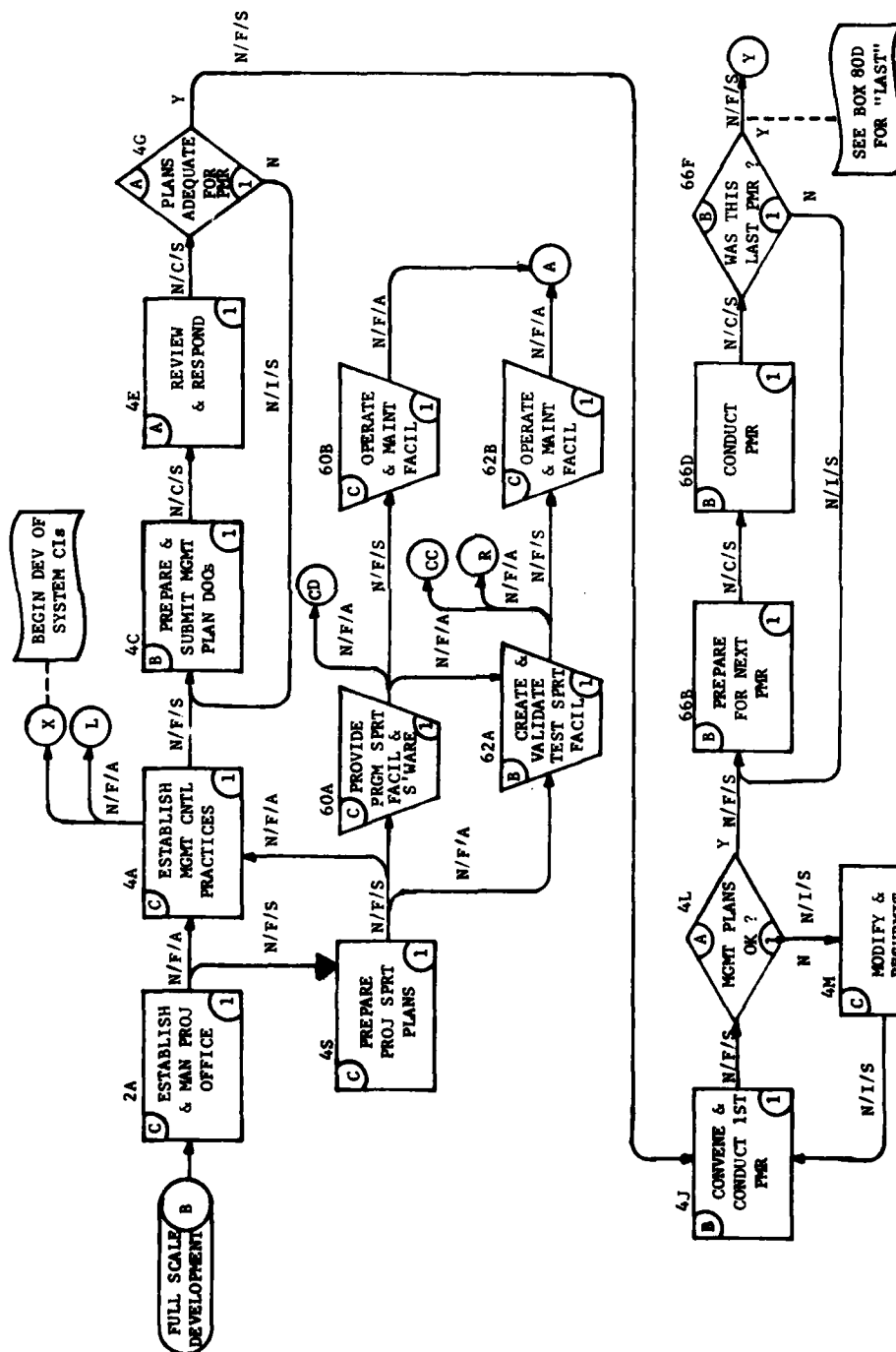
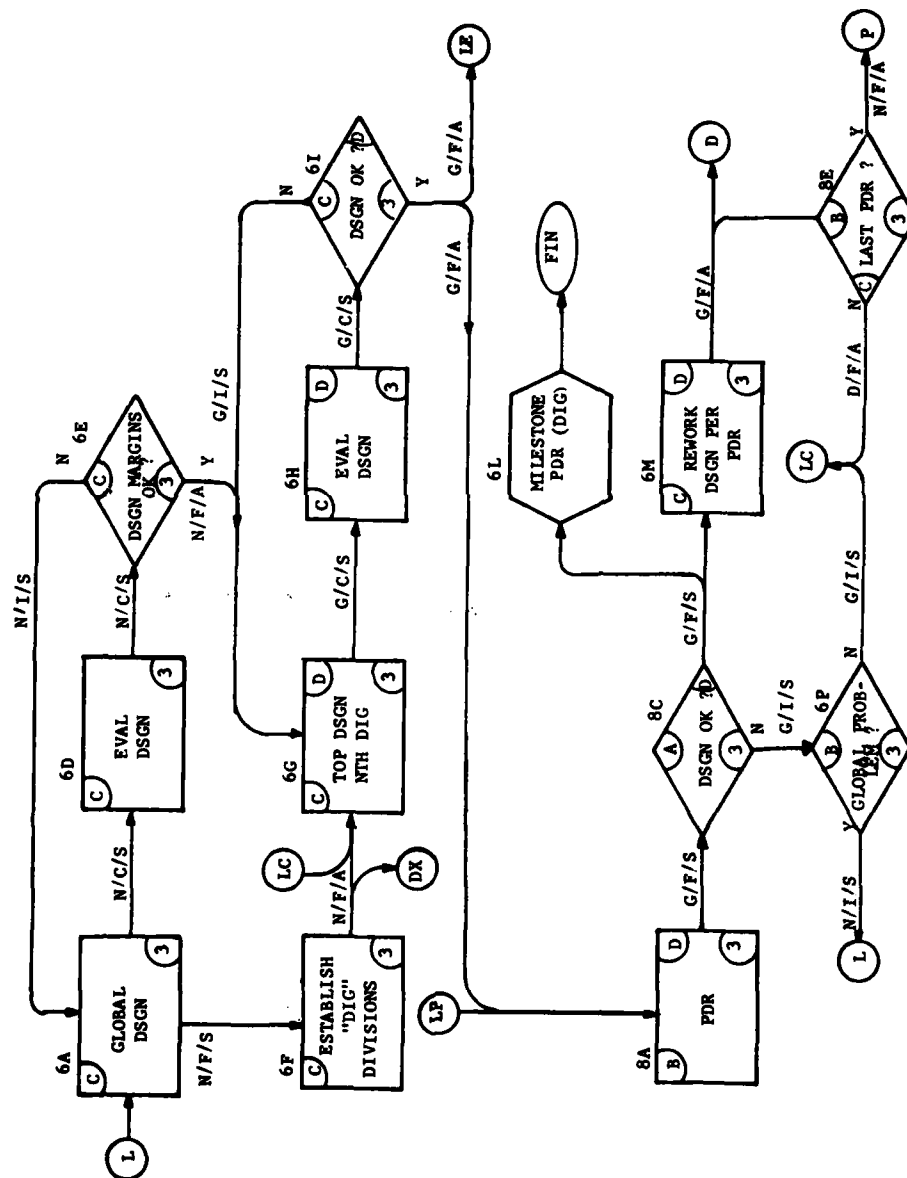


Figure A-2. Software Acquisition Process Flow Diagram - LoSim Level  
Sheet 1 - Organization, Management, Support



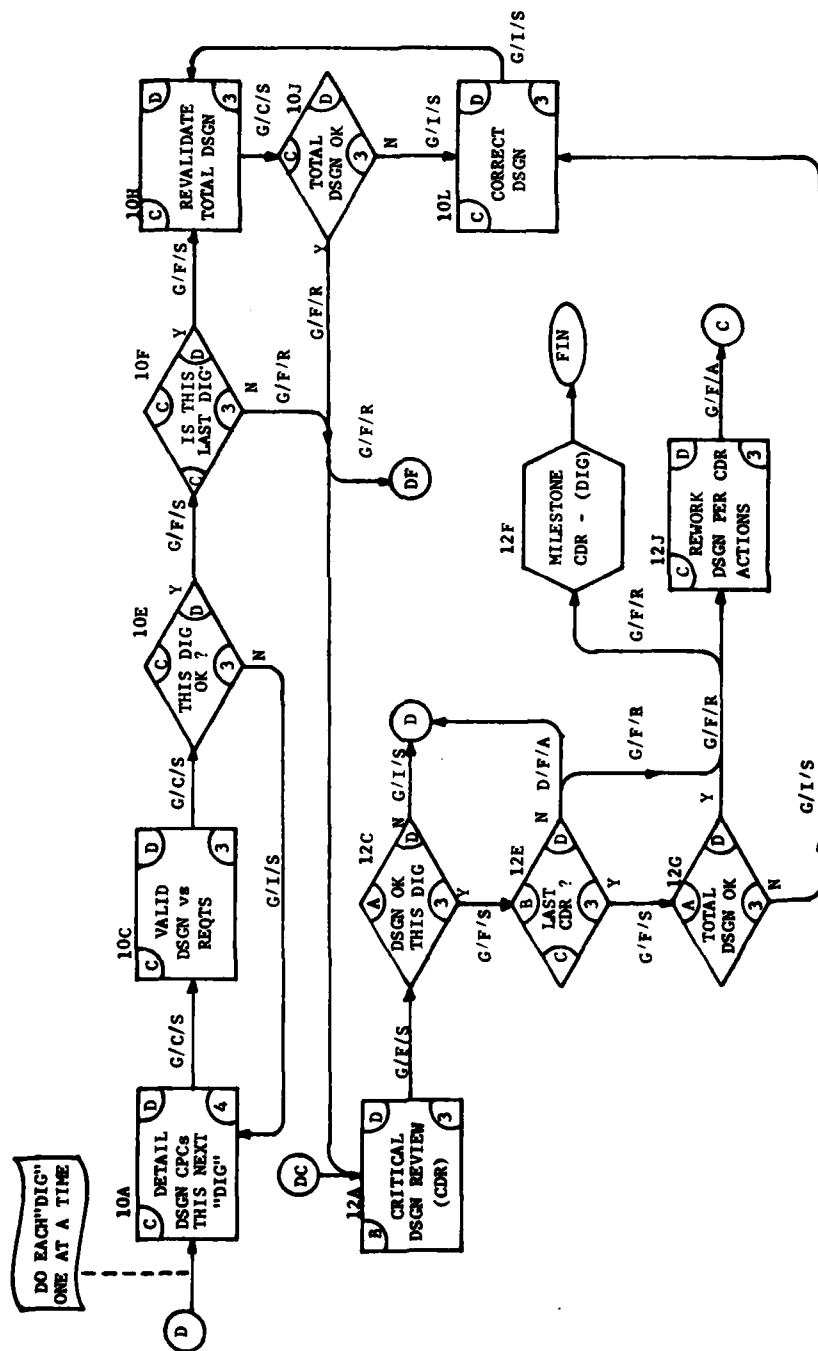
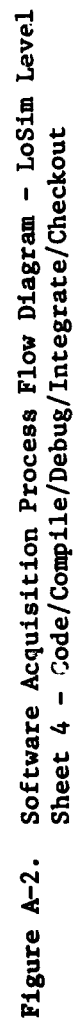


Figure A-2. Software Acquisition Process Flow Diagram - LoSim Level  
Sheet 3 - Detail Design Thru CDR



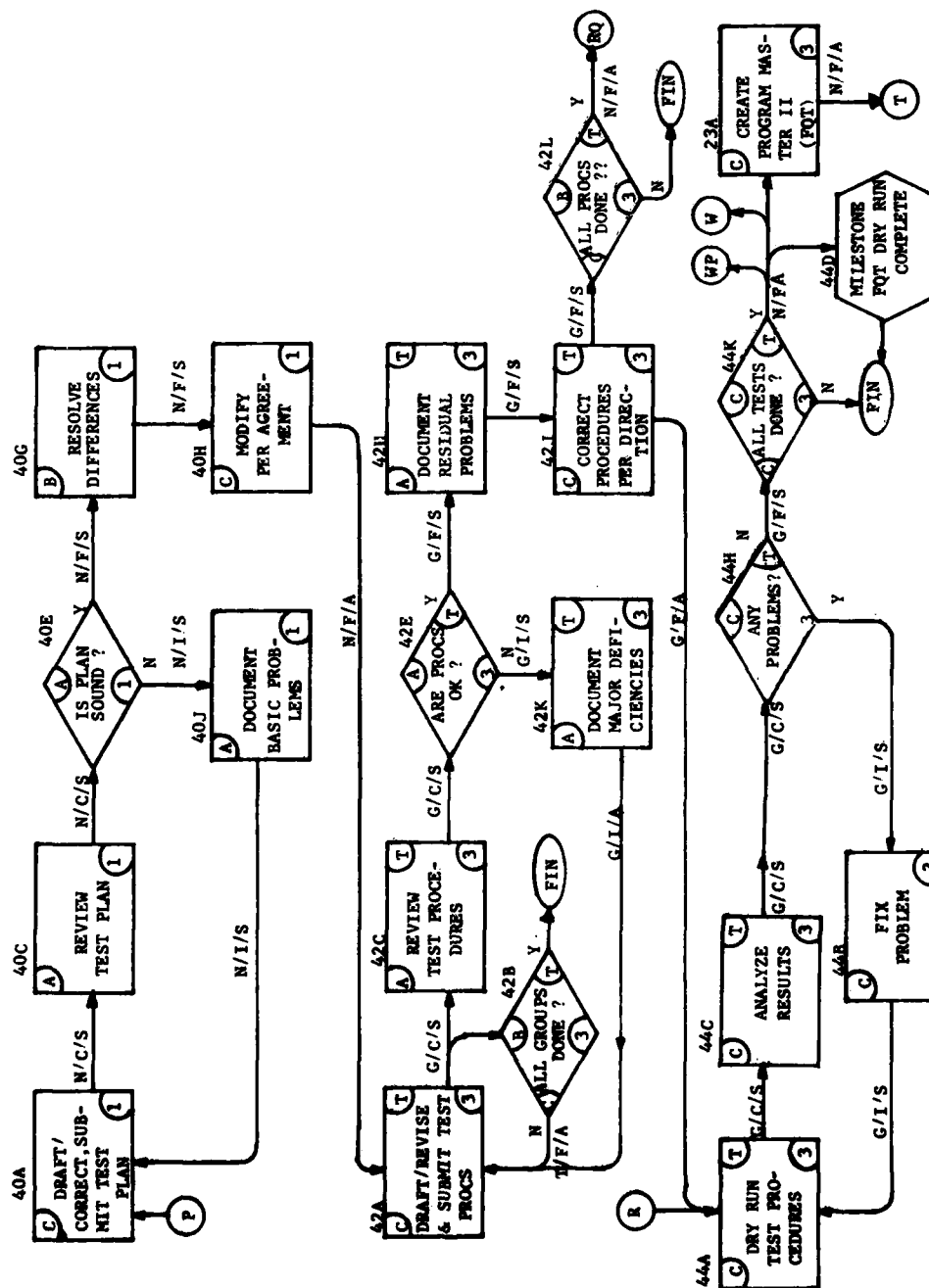
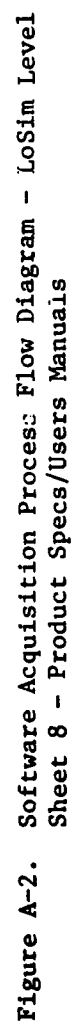


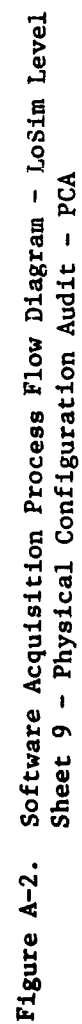
Figure A-2. Software Acquisition Process Flow Diagram - LoSim Level  
Sheet 5 - Test Plan/Procedures/Dry Runs











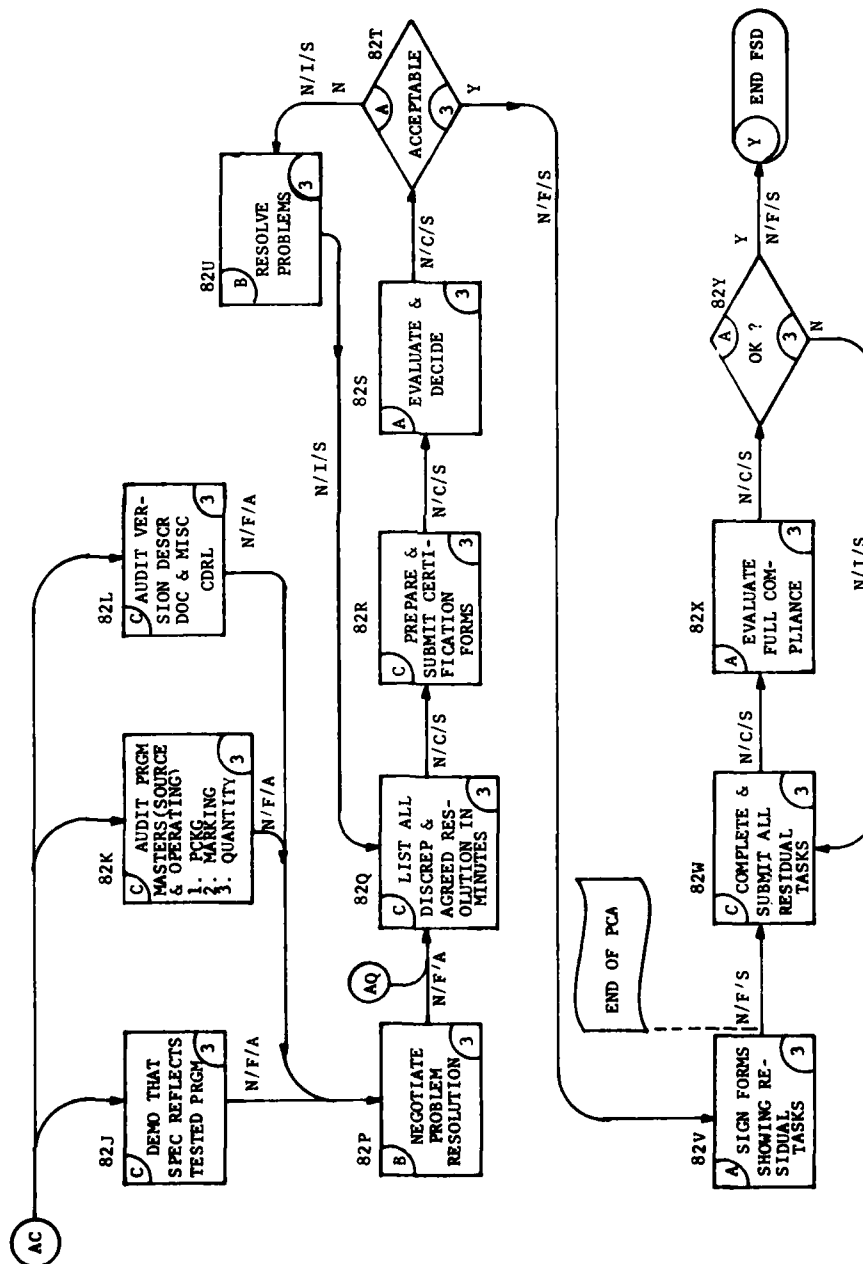


Figure A-2. Software Acquisition Process Flow Diagram - LoSim Level  
Sheet 10 - Physical Configuration Audit - PCA (Concluded)



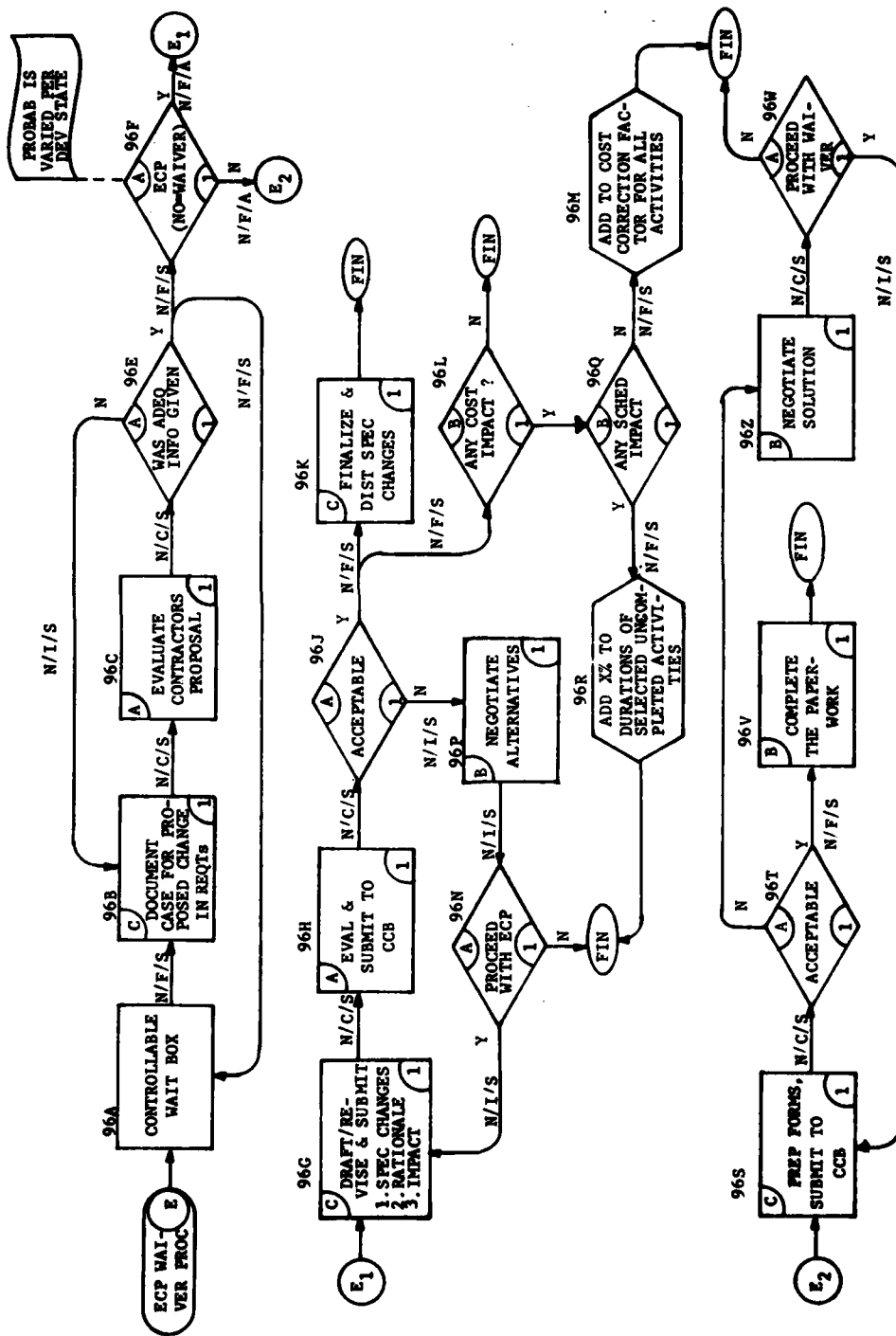


Figure A-2. Software Acquisition Process Flow Diagram - LoSim Level  
Sheet 12 - ECP Subprocess

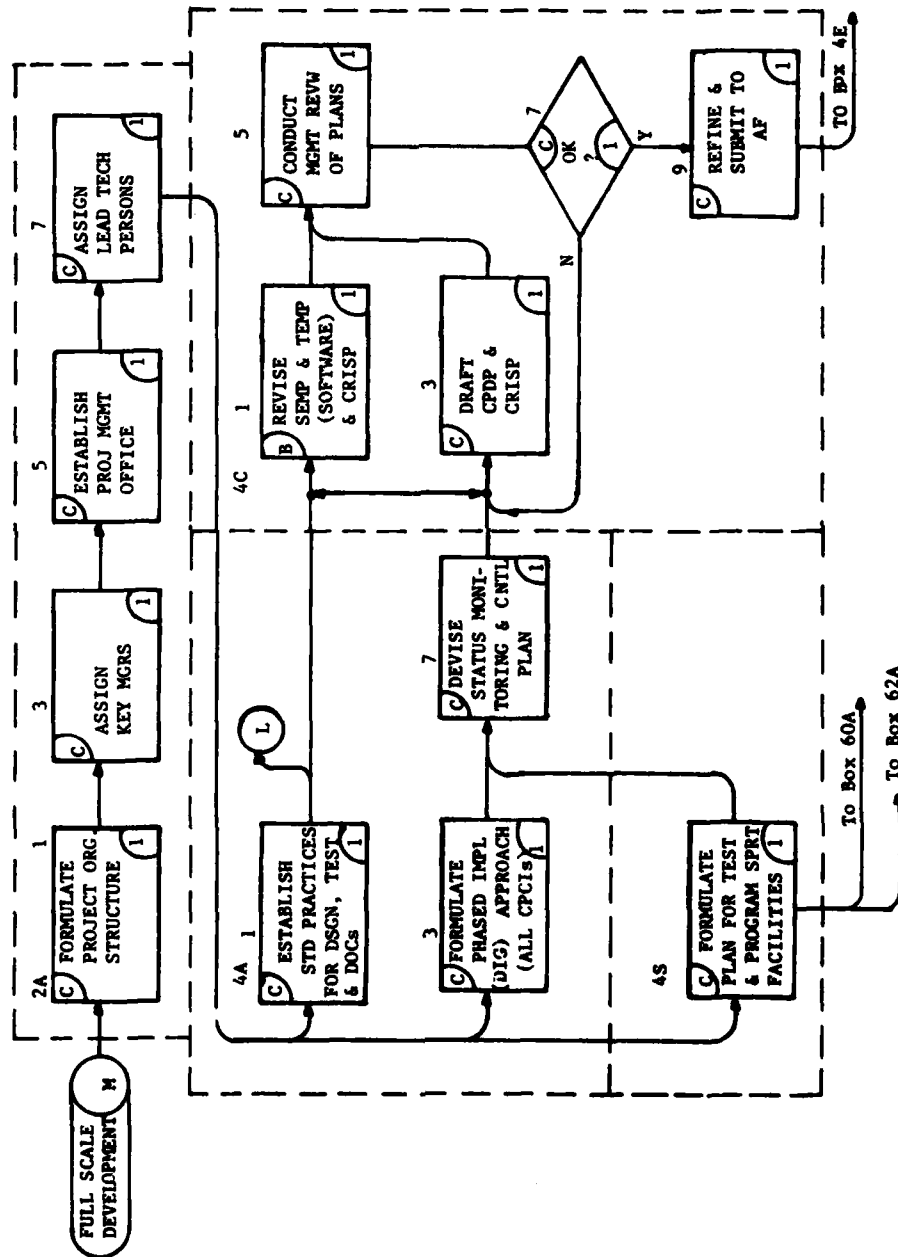


Figure A-3. Process Flow Expansion  
Sheet 1 - Boxes 2A, 4A, 4C & 4S

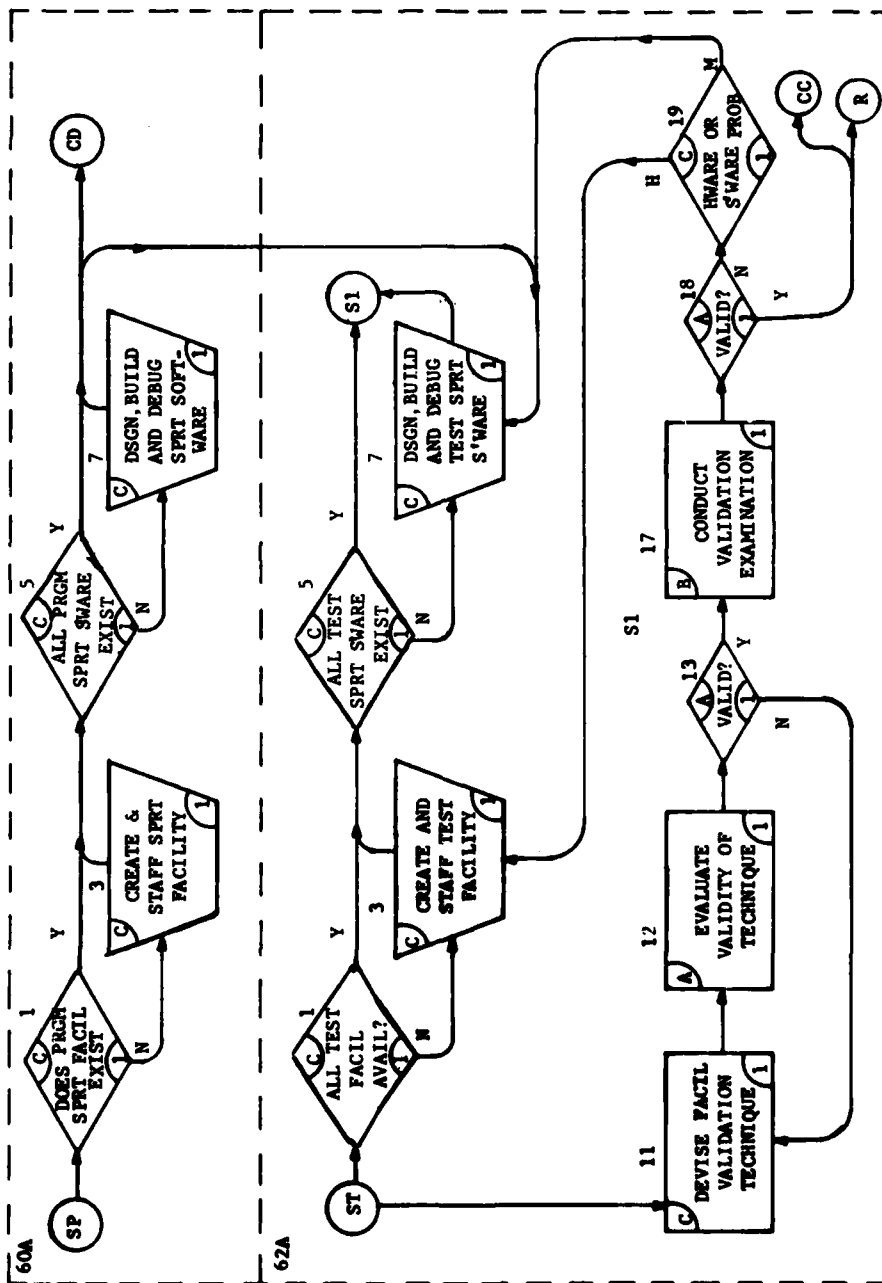


Figure A-3. Process Flow Expansion  
Sheet 2 - Boxes 60A & 62A





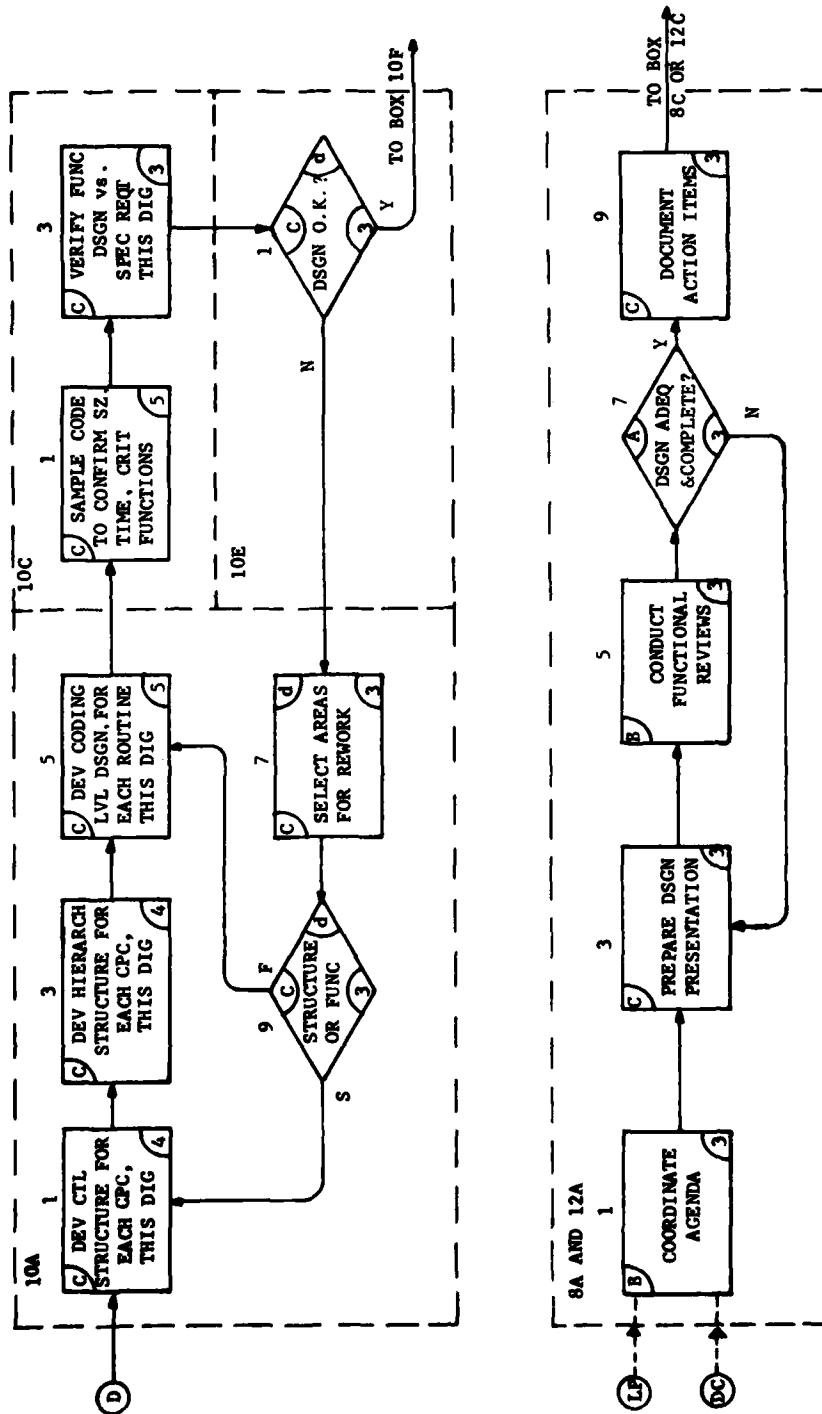


Figure A-3. Process Flow Expansion  
Sheet 4 - Boxes 8A or 12A, 10A, 10C & 10E

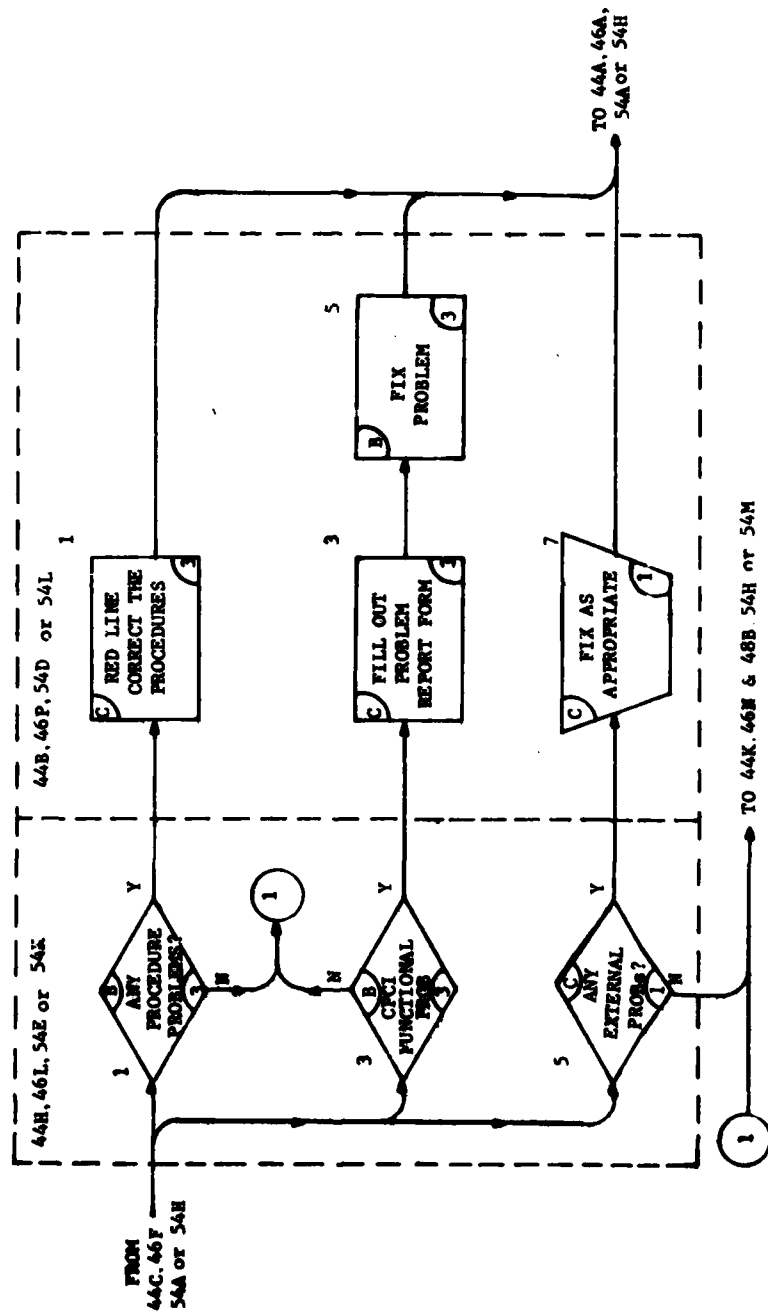


Figure A-3. Process Flow Expansion  
Sheet 5 - Boxes 44H, 44B, 46L, 46P, 54E, 54D, 54K, 54L

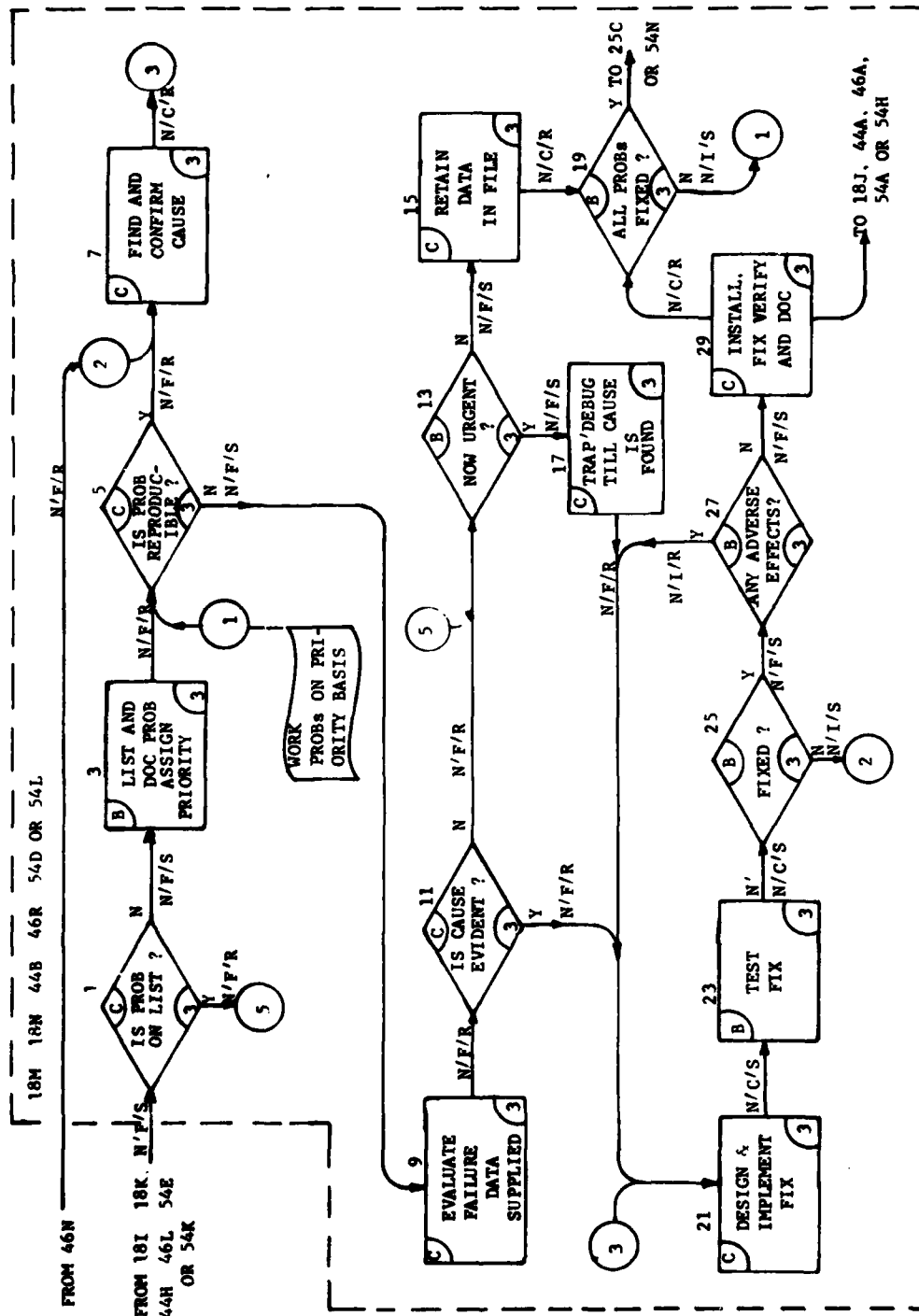


Figure A-3. Process Flow Expansion  
Sheet 6 - Boxes 18M, 18N, 46R, 54D, or 54L

Table A-2

## PROCESS FLOW DIAGRAM AMPLIFICATION NOTES

<u>Note No.</u>	<u>Fig. A-2 Sht.</u>	<u>Fig. A-2 Box</u>	<u>Fig. A-3 Sht.</u>	<u>Fig. A-3 Box</u>	<u>Amplification Notes</u>
1	1	2A	1	2A3 2A7	The assignment of key personnel at the initiation of a project is generally a slow process. Each person selected for a new project usually has an existing assignment which must be transitioned to a successor; the successor may also need to transition his job to another, etc. Advance planning by the contractor helps in the personnel selection process but the uncertainties associated with the award and timing on this contract (as well as on other contracts bid) make startup a traumatic event that gets under way slowly.
2	1	4A	1	4A3	Just the concept and general approach to the Developmental Integration Group (DIG) (see Section 4.1.5) plan are established here to provide a basis for the status monitoring and management plans. The grouping of specific CPCs into DIGs is established in Box 6F. (See Note 11).  Note that both 4A3 and 4S support activity 4C3 (CPDP preparation) even though the direct connection isn't shown on the LoSim diagram. This feedback is shown indirectly via the 4S to 4A to 4C connection; this arrangement will satisfy the precedence needs of the Simulator.
3	1	4C	1	4C1	The System Engineering Management Plan (SEMP), the Test and Evaluation Management Plan (TEMP), and the Computer Resources Integrated Support Plan (CRISP) are normally prepared during the system's Validation Phase. These plans usually need updating in the light of the current contract and contractor. This box covers only those portions concerned with software.
4	1	4C	1	4C3	The Computer Program Development Plan (CPDP) is generally addressed in the contractor's proposal. This activity covers the rewrite and extension necessary before this plan can be put into effect contractually.

Table A-2 (Continued)

<u>Note No.</u>	<u>Fig. A-2 Sht. Box</u>	<u>Fig. A-3 Sht. Box</u>	<u>Amplification Notes</u>
5	1 4S	1 4S	Per Section 4.1.4 this activity provides for the most general case where the program and test support facilities are not identical, even though some portions of the hardware may be shared for both uses.
6	1 60A	2 60A1-60A7	A need to build support software will significantly increase this activity's elapsed time over that otherwise required. Parameter values for this activity must be selected to reflect the actual (or expected) contractual situation.
7	1 62A	2 62A11 62A12 62A13 62A17 62A18 62A19	Any non-trivial special (i.e., not specified) equipment or software which is to be used to support Qualification Testing must be evaluated to assure that it is valid for its intended use. As examples, a facility may be needed to emulate a non-available interfacing component (hardware or software) or to produce radar returns representing a flying aircraft, etc. Any deliverable test support component would not be processed in these boxes because its validity would be established in the tests associated with its acceptance by the government.
8	1 4G 4J 4L 4M		The management plans are frequently resolved at the first full-scale overall Program Management Review (PMR), as shown. Instead, they can be treated at a separate meeting (if they become urgent issues), or without a meeting (by mail and phone) if not controversial; the process parameters can be adjusted to cover any expected case.
9	1 66B 66D 66F 9 80D		PMRs are generally conducted on a periodic basis (e.g., monthly or bimonthly) throughout the entire contractual period. They are shown here because the preparation and conduct activities consume considerable manpower on an intermittent basis and thereby can impact the development process. Note that a Special Event Box (80D) will cause the PMR activity to stop at the start of PCA.

Table A-2 (Continued)

<u>Note No.</u>	<u>Fig. A-2 Sht. Box</u>	<u>Fig. A-3 Sht. Box</u>	<u>Amplification Notes</u>
10	2 6A 6D	3 All	The design and evaluation activities shown are representative of those conducted on many projects; they are not intended as an all inclusive set. In general, the overall design is sampled at a moderate depth while design areas that are perceived to be risky, difficult, or innovative are given emphasis.
11	2 6F		Here the specific Developmental Integration Groups (see Section 4.1.5) comprising the CPCI are defined.
12	2 6G 6H 6I		The design activities conducted prior to these boxes are global in that they include the overall CPCI at a fairly gross level. They establish that the overall system concept is feasible and can accord with space, timing, and other restrictions. In these boxes, the capabilities to be provided in each DIG (see Section 4.1.5) are designed to a depth necessary to show that the approach for each specific function is feasible.
13	2 6I	4 8A1- 8A9	Once the contractor establishes the adequacy of his design (in box 6I) he must document it (using Product Specification format) and submit it for government review and approval. This is shown via connectors LE and LC to Boxes 20A to 20E on Sheet 8 of Figure A-2.
14	2 6M 8C 12A 12J		Even when the PDR results (in Box 8C) are satisfactory, there are generally a number of specific deficiency areas noted during the extensive review. These are documented by the contractor in the design review minutes as items which he agrees to correct; Box 6M makes provision for these corrections. The references to Boxes 12A and 12J indicate that this note also applies to the CDR results.
15	2 8E 10F 12E 12H		Each of the Decision Boxes branches on a count rather than on the basis of probability. If the design at these points has not been completed for all the DIGs, this causes the design process to repeat, but on

Table A-2 (Concluded)

<u>Note</u> <u>No.</u>	<u>Fig. A-2</u> <u>Sht. Box</u>	<u>Fig. A-3</u> <u>Sht. Box</u>	<u>Amplification Notes</u>
	18H		the next DIG; e.g., at Box 6G. Note that regardless of the counter, further design on the current DIG continues; e.g., by transfer to connector D.
	18P		
	42B		
	42L		
	44K		
	46N		
	48D		



## Appendix A Abbreviations

AF	Air Force
ADEQ	Adequate
CCB	Configuration Control Board
CCI&C	Code, Compile, Integrate & Check
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CI	Configuration Item
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPDP	Computer Program Development Plan
CPT&E	Computer Program Test & Evaluation
CRISP	Computer Resources Integrated Support Plan
CRIT	Critical
CTL	Control
DEMO	Demonstrate
DESCR	Description
DEV	Develop
DIG	Developmental Integration Group
DISCREP	Discrepancies
DIST	Distribute
DOC	Document
DSGN	Design
ECP	Engineering Change Proposal
EVAL	Evaluate

## Appendix A Abbreviations (Continued)

FACIL	Facility
FCA	Functional Configuration Audit
FIN	End of this process flow diagram path
FQT	Formal Qualification Testing
FUNC	Functional
HIERARCH	Hierarchial
HWARE	Hardware
I&C	Integration and Checkout
IMPL	Implementation
INFO	Information
INTEG	Integration
LVL	Level
MAINT	Maintain
MGMT	Management
MGR	Manager
MISC	Miscellaneous
ORG	Organization
PCA	Physical Configuration Audit
PCKG	Packaging
PDR	Preliminary Design Review
PRGM	Program
PMR	Program Management Review
PREP	Prepare

## Appendix A Abbreviations (Concluded)

PROB	Problem
PROC	Procedure
PROD	Product
PROG	Programming
PROJ	Project
REQT	Requirement
REVAL	Reevaluation
REVV	Review
SCHED	Schedule
SEMP	System Engineering Management Plan
S'WARE	Software
SPEC	Specification
SPRT	Support
STD	Standard
SYS	System
SZ	Size
TECH	Technical
TEMP	Test and Evaluation Master Plan
TIG	Test Integration Group

## APPENDIX B

### MODEL DEFINITION DATA

This appendix incorporates and describes the tables that define the software acquisition process logic and parameter values to the Simulator. These tables are input via a terminal to computer files and may easily be altered. As explained in Section 6.4, the Simulator reads these files, reformats the tables, and interprets the revisions to develop simulation results. Within broadly-defined limits the tables may be modified, to represent more or less detail, differences in process logic, or revised parameter values. Without needing revision itself, the Simulator will interpret the modified tables and develop corresponding simulation results.

Table B-1, Software Acquisition Process Model Network Linkage, is a tabular representation of the entire LoSim Process Flow Diagram (Figure A-2). Table B-2, Software Acquisition Process Model Activity Box Parameter Data, contains the manning and duration parameter value estimates thus far developed for the activities depicted in the LoSim Process Flow Diagram. Table B-3, Software Acquisition Process Model Decision Box Parameter Data, contains estimates of the decision outcome probabilities for all LoSim Flow Diagram Decision Boxes except counters. Table B-4, Software Acquisition Process Model Counter & Special Event Box Parameter Data, contains the LoSim Flow Diagram counter Decision Box limits and Special Event Box parameters so far defined. The latter are mainly Milestone identifiers.

The columns of these tables, and the values that the data in each column may legitimately contain, are explained below.

#### 1. TABLE B-1

Table B-1 represents the Process Model network. It must contain an entry for each box in the Process Flow Diagram that it represents. There is currently an entry in Table B-1 for every box in the LoSim Process Flow Diagram (Figure A-2).

##### 1.1 Box Data

a. Box ID: This is the box's label (see Figure A-1).

b. Box Type:

A = a mainstream Activity Box.

B = a branching box (i.e., a normal Decision Box).

C = a counter Decision Box. This is similar to a type B box, except that the exit is determined by whether an incrementing counter has reached it's limit; see Table A-2, Note 15.

E = a Special Event Box. This provides for special actions at designated locations in the process flow. These include displaying Milestones, resetting counters, and changing parameter values, and provide for as yet undefined future needs.

H = a helping box (i.e., a support Activity Box). See Figure A-1.

c. Box Lns: This is the number of Table B-1 lines used to define this box. This field is included as a programming convenience. Multiple lines are needed to accommodate multiple predecessors or successors within the table format.

## 1.2 General Data

a. Doer: This defines the agency or agencies assigned to perform the activity or to make the decision:

A = Government (e.g., Air Force)

C = Contractor

B = Both.

b. Lvl: This defines the system component level at which the activity is performed or the decision is made, as listed below. Note that a box at any level may also be divided (or clustered) to conform with Integration Group assignments, described subsequently.

1 = System level

2 = Segment level (see Section 4.1.2)

3 = CPCI level

4 = CPC level

5 = Computer Program Routine level, or lower.

c. Box Grp: This defines the box's membership (if any) within an Integration Group (see Section 4.1.5).

D = Developmental Integration Group (DIG)

T = Test Integration Group (TIG)

N = No Integration Group.

d. Subnet: A user may assign the box to any one of up to 15 Subnetworks by entering a number in the range 1-15 in this column. The Simulator will develop aggregate timing and cost data for each Subnetwork as well as the entire network.

### 1.3 Predecessors

a. **Quan:** This specifies the quantity of the boxes' immediate predecessors. The data required for the first or only predecessor are specified in the "Predecessors" columns on this line. If a box has more than one predecessor, the data for its second and any subsequent predecessors are stored in corresponding columns of successive lines. The "Quan" column of these successive lines is blank.

b. **Box:** This is the Box ID (see paragraph 1.1a) of the predecessor.

c. **Exit:** This is the predecessor box's exit used to reach this box:

Y = "Yes" exit

N = "No" exit

S = Single exit.

d. **Grp:** The box's Group Control parameter, used to maintain Group (i.e., DIG or TIG) number continuity and incrementation during network flow.

N = No group involvement

D = Increment DIG Number

T = Increment TIG number

G = Retain predecessor's group number.

e. **PM:** A parameter used to indicate Progression Mode during box-to-box progression. See Figure A-1.

F = Normal forward progression

I = Iterative progression

C = Continue Progression Mode of predecessor.

f. **Strt:** Defines the combination of predecessors that must finish before this box may start.

A = "AND" relationship. This predecessor's completion is a necessary but not a sufficient condition for starting the box.

R = "OR" relationship. Completion of only one type R predecessor is necessary to start the box. Predecessors of other types, if specified, are also required.

S = Start immediately. This predecessor's completion by itself is sufficient to start this box.

**Note:** If a box has no predecessor (e.g., Box 2A) all of its entry's predecessor fields are zero filled.

#### 1.4 Successors

a. Qnys: This specifies the quantity of the box's "Yes" exit (or single exit) immediate successors.

b. Qnn: This specifies the quantity of the box's "No" exit immediate successors.

c. Ysbx: "Yes" successor Box ID. If a box has more than one "Yes" successor, their box IDs are stated in successive lines in this column.

d. Nbx: "No" successor Box ID. If a box has more than one "No" successor, their box IDs are stated in successive lines in this column. Note that Special Event Boxes may use the Nbx column to list the IDs of remote boxes which are targeted for change.

#### 2. TABLE B-2

Table B-2 contains the parameter data for each Activity Box (box types A & H) in Table B-1. Every Activity Box must have a Table B-2 entry. Tables B-3 & B-4 contain the parameter data for the other box types.

##### 2.1 Box Data

a. Box ID: This is the box's label, which must be identical to its Table B-1 Box ID (see paragraph 1.1a).

b. Box Type: This must be the same as the Table B-1 entry's Box Type (see paragraph 1.1b).

c. Box Grp: Identical to the Table B-1 entry's Box Grp (see paragraph 1.2c).

##### 2.2 Manpower

Manpower is subdivided into five categories of work for contractor personnel, and three for government personnel, as explained below. Note that management personnel are not assigned to specific activities. Instead, manpower and dollar costs representing a given management structure will be sustained for the project as a whole, or for designated parts of it. Management personnel effort is not shown for specific boxes even if the work is largely done by such persons.

The table contains a pair of columns for each manpower category; i.e.:

a. Contractor

Sys = System engineers and analysts

Dsgn = Designers (junior and senior)

Prgm = Computer programmers

Test = Software test engineers

Sprt = Support personnel; e.g., writers, operators,  
maintenance persons.

b. Government

Dev = Developing Command (e.g. ESD)

Usr = Using Command (e.g. TAC)

Sprt = Supporting Command (e.g., AFLC).

For each manpower category the first column contains the (integral) number of personnel of that type estimated necessary to complete the activity. The second column allows assignment to be on a part-time basis; i.e., by showing the proportion of their time (in tenths) these personnel apply to this task. Here blank = full time.

c. Iterate Factor:

Many tasks may need to be repeated because the results achieved on the first pass were not adequate to meet subsequent needs or review criteria. Since the work required on subsequent passes usually involves fewer persons, these three columns each contain a factor (from 0 to 10) representing the number of tenths by which the original number of persons in each of the manpower columns (as specified for the first pass) must be multiplied to obtain the manpower needed respectively on the second, third, and fourth or later iteration of the activity.

### 2.3 Durations

a. Days: The first duration column contains the mean duration of the activity, in work days.

b. It Fctr: The next three columns each contain a factor (from 0 to 10) representing the number of tenths of the first iteration's duration (i.e., days column) required to complete the second, third, and fourth or later iteration, respectively.



#### 2.4 Wait

This field may contain a waiting time (in days) before the activity may begin. The action may begin only after the wait period has completed; the Wait itself starts after all predecessor conditions are satisfied. If blank, no Wait is required.

#### 2.5 Notes

This column refers to the notes listed within Table A-2, Process Flow Diagram Amplification Notes.

### 3. TABLE B-3

Each Table B-3 entry contains the parameter data for a normal Decision Box (box type B) with an entry in Table B-1. Every normal Decision Box in Table B-1 must be represented by a Table B-4 entry.

#### 3.1 Box Data

These fields' definitions are given in paragraphs 2.1a-c.

#### 3.2 Yes Exit Probability

These four columns contain the probabilities (each multiplied by 100) of taking the "Yes" exit on the first four iterative passes through the Decision Box; see paragraph 2.2c. The leftmost column provides first pass probability. The rightmost column probability will be used repeatedly, if the box is iterated more often than four times.

#### 3.3 Wait

See paragraph 2.4.

#### 3.4 Notes

See paragraph 2.5.

### 4. TABLE B-4

This table contains an entry for each counter Decision Box (type C) and each Special Event Box defined in Table B-1. Every such box must be represented by a Table B-4 entry.

#### 4.1 Box Data

These fields' functions are given in paragraph 2.1a-c.

#### 4.2 Funct

Three types of function have thus far been allocated to Special Event Boxes:

M = Milestone. The contents of the Event Label column (a Milestone name) will be output on schedule reports for each Special Event Box entered. Where "(DTN.NO)" is included in the message the Integration Group number at the time of entry into the box will be printed as part of this output.

R = Reset. The stored data and status of the boxes listed in the Table B-1 Nbx column will be reset for any Integration Group (DIG or TIG) in effect as of the time the box was entered.

P = Parameter modification. The parameters given in the Parameter column will replace the existing values for the boxes listed in the Table B-1 Nbx column.

#### 4.3 Event Label

This contains the characters to be output as the Milestone name for a Milestone-type Special Event Box.

#### 4.4 Parameter

This column identifies the parameter which is to be changed by a reset (type R) or parameter modification (type P) Special Event.

#### 4.5 Notes

See paragraph 2.5.

Table B-1 Software Acquisition Process Model  
Network Linkage

----BOX-----						---GENERAL DATA---				-----PREDECESSORS-----				----SUCCESSORS-----			
ID	Type	Lns	Doer	Lvl	Grp	Sub Net	Quan	Box	Exit	Grp	PM	Strt	QNYS	QNN	YSBX	NBX	
2A	A	2	C	1	N		0	0	0	0	0	0	2	0	4A 4S		
4A	A	4	C	1	N		2	2A 4S	S	N	F	A	4	0	4C 6A 53A 53C		
4C	A	2	B	1	N		2	4A	S	N	F	S	1	0	4E		
4E	A	1	A	1	N		1	4G	N	N	I	S	1	0	4G		
4G	B	1	A	1	N		1	4E	S	N	C	S	1	1	4J	4C	
4J	A	2	B	1	N		2	4G	Y	N	F	S	1	0	4L		
4L	B	1	A	1	N		1	4M	S	N	I	S	1	1	66B 4M		
4M	A	1	C	1	N		1	4J	N	N	I	S	1	0	4J		
4S	A	3	C	1	N		1	2A	S	N	F	S	3	0	4A 60A 62A		
6A	A	3	C	3	N		3	4A 6P	S Y	N N	F I	A S	2	0	6D 6F		
6D	A	1	C	3	N		1	6A	S	N	C	S	1	0	6E	6A	
6E	B	1	C	3	N		1	6D	S	N	C	S	1	1	6G		
6F	A	2	C	3	N		1	6A	S	N	F	S	2	0	6G 14A		
6G	A	5	C	3	D		5	6F 6E	S Y	N N	F F	A A	1	0	6H		
6H	A	1	C	3	D		1	6I 6P 8E	N N N	G G D	I I F	S S A	1	0	6I		

Table B-1 (Continued)

-----BOX-----				--GENERAL DATA---				Sub	-----PREDECESSORS-----							----SUCCESSORS-----			
ID	Type	Lns	Doer	Lvl	Grp	Net	Quan	Box	Exit	Grp	PM	Strt	QNYS	QNN	YSBX	NBX			
6I	B	2	C	3	D		1	6H	S	G	C	S	2	1	8A	6G			
6L	E	1	N	3	N		1	8C	Y	G	F	S	0	0	20A				
6M	A	2	C	3	D		1	8C	Y	G	F	S	2	0	99F				
6P	B	1	B	3	N		1	8C	N	G	I	S	1	1	8E	10A			
															6A	6G			
8A	A	2	B	3	D		2	6I	Y	G	F	A	1	0	8C				
8C	B	2	A	3	D		1	20E	Y	G	F	A	2	1	6L	6P			
								8A	S	G	F	S			6M				
8E	C	1	B	3	D		1	6M	S	G	F	A	1	1	40A	6G			
10A	A	4	C	4	D		4	6M	S	G	F	A	1	0	10C				
								12C	N	G	I	S							
								12E	N	D	F	A							
								10E	N	G	I	S							
10C	A	1	C	3	D		1	10A	S	G	C	S	1	0	10E				
10E	B	1	C	3	D		1	10C	S	G	C	S	1	1	10F	10A			
10F	C	2	C	3	D		1	10E	Y	G	F	S	1	2	10H	12A			
															24A				
10H	A	2	C	3	D		2	10F	Y	G	F	S	1	0	10J				
								10L	S	G	I	S							
10J	B	2	C	3	D		1	10H	S	G	C	S	2	1	12A	10L			
															24A				
10L	A	2	C	3	D		2	10J	N	G	I	S	1	0	10H				
								12G	N	G	I	S							
12A	A	3	B	3	D		3	10F	N	G	F	R	1	0	12C				
								10J	Y	G	F	R							
								24E	Y	G	F	A							
12C	B	1	A	3	D		1	12A	S	G	F	S	1	1	12E	10A			

Table B-1 (Continued)

----BOX-----						---GENERAL DATA---						-----PREDECESSORS-----						---SUCCESSORS---					
ID	Type	Lns	Doer	Lvl	Grp	Sub Net	Quan	Box	Exit	Grp	PM	Strt	QNYs	QNN	YSEX	NBX							
12E	C	3	B	3	D		1	12C	Y	G	F	S	1	3	12G	10A 12F 12J							
12F	E	2	N	3	N		2	12E	N	G	F	R	0	0	99F								
12H	C	2	B	3	D		1	12G	Y	G	F	R	2	0	26A	99F							
12J	A	2	C	3	D		2	24E	Y	N	F	S	1	0	70A								
								12E	N	G	F	R	1	0	16A								
								12G	Y	G	F	R											
14A	A	2	C	3	N		1	6F	S	N	F	A	2	0	14C								
14C	A	1	C	3	N		1	14A	S	N	F	S	1	0	18J								
16A	A	2	C	5	D		2	12J	S	G	F	A	1	0	16C								
16C	A	2	C	5	D		2	18H	N	D	F	A	1	0	18C								
								16A	S	G	F	A	1	0									
								60A	S	N	F	A											
18C	A	2	C	5	D		2	16C	S	G	F	A	1	0	18E								
								14A	S	N	F	A											
18E	A	1	C	3	D		1	18C	S	G	F	S	1	0	18G								
18G	A	3	C	3	D		1	18E	S	G	F	S	3	0	18H								
															18J								
															18R								
18H	C	1	C	3	D		1	18G	S	G	F	S	0	1	99F	16A							
18I	B	1	C	3	D		1	18F	N	G	C	S	1	1	18M	18K							

Table B-1 (Continued)

---BOX-----				--GENERAL DATA---				Sub				-----PREDECESSORS-----				---SUCCESSORS---			
ID	Type	Lns	Doer	Lvl	Grp	Net	Quan	Box	Exit	Grp	PM	Strt	QNYS	QNN	YSBX	NBX			
18J	A	8	C	3	D		8	62A	S	N	F	A	1	0	18F				
								14C	S	G	F	A							
								18G	S	G	F	A							
								18P	N	D	F	A							
								62C	S	G	I	S							
								18M	S	G	I	S							
								18N	S	G	I	S							
								18Q	S	G	I	S							
18K	B	1	C	3	D		1	18I	N	G	C	S	1	1	18N	18L			
18L	B	1	C	3	D		1	18K	N	G	C	S	1	1	18Q	18P			
18F	B	1	C	3	D		1	18J	S	G	C	S	1	1	62C	18I			
18M	A	1	C	3	D		1	18I	Y	G	I	S	1	0	18J				
18N	A	1	C	3	D		1	18K	Y	G	I	S	1	0	18J				
18P	C	1	C	3	D		1	18L	N	N	F	S	1	1	22A	18J			
18Q	A	1	C	3	D		1	18L	Y	G	I	S	1	0	18J				
18R	E	1	N	3	N		1	18G	N	G	F	S	0	0	99F				
20A	A	2	C	3	D		2	20E	N	N	I	S	1	0	20C				
								6I	Y	G	F	A							
20C	A	1	A	3	D		1	20A	S	G	C	S	1	0	20E				
20E	B	2	A	3	D		1	20C	S	G	C	S	2	1	24A	20A			
															8A				
22A	A	1	C	3	N		1	18P	Y	N	F	S	1	0	44A				
23A	A	1	C	3	N		1	44K	Y	N	F	S	1	0	46A				
24A	A	4	C	3	D		4	20E	Y	G	F	A	1	0	24C				
								24E	N	G	I	S							
								10F	N	G	F	R							
								10J	Y	G	F	R							
24C	A	1	A	3	D		1	24A	S	G	C	S	1	0	24E				

Table B-1 (Continued)

----BOX-----				--GENERAL DATA--			Sub	-----PREDECESSORS-----					----SUCCESSORS-----				
ID	Type	Lns	Doer	Lvl	Grp	Net	Quan	Box	Exit	Grp	PM	Strt	QNYs	QNN	YSBX	NBX	
24E	B	2	A	3	D		1	24C	S	G	C	S	2	1	12H 12A	24A	
25C	A	1	C	3	N		1	46R	S	N	C	S	1	0	46S		
26A	A	5	C	3	N		2	12H 44K	Y Y	N N	F F	A A	5	0	28A 80A 80C 80E 80D		
28A	A	2	C	3	N		2	26A 46T	S Y	N N	F F	A A	1	0	82E		
40A	A	2	C	1	N		2	8E 40J	Y S	N N	F I	A S	1	0	40C		
40C	A	1	A	1	N		1	40A	S	N	C	S	1	0	40E	40J	
40E	B	1	A	1	N		1	40C	S	N	C	S	1	1	40G		
40G	A	1	B	1	N		1	40E	Y	N	F	S	1	0	40H		
40H	A	1	C	1	N		1	40G	S	N	F	S	1	0	42A		
40J	A	1	A	1	N		1	40E	N	N	I	S	1	0	40A		
42A	A	3	C	3	T		3	40H 42B	S N	N T	F F	A A	2	0	42C 42B		
42B	C	1	B	3	T		1	42A	S	G	C	S	0	1	99F	42A	
42C	A	1	A	3	T		1	42A	S	G	C	S	1	0	42E		
42E	B	1	A	3	T		1	42C	S	G	C	S	1	1	42H	42K	
42H	A	1	A	3	T		1	42E	Y	G	F	S	1	0	42J		
42J	A	2	C	3	T		1	42H	S	G	F	S	2	0	44A 42L		
42K	A	1	A	3	T		1	42E	N	G	I	S	1	0	42A		
42L	C	1	B	3	T		1	42J	S	G	F	S	1	0	50A	99F	

Table B-1 (Continued)

----BOX-----   --GENERAL DATA---   -----PREDECESSORS-----   ----SUCCESSORS-----																	
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QMS	QNN	YSBX	NBX
42M	A	1	B	3	N			1	46N	Y	N	F	A	1	0	42N	
42N	A	1	C	3	N			1	42M	S	N	F	S	1	0	46S	
44A	A	4	C	3	T			4	42J	S	G	F	A	1	0	44C	
									22A	S	N	F	A				
									62A	S	N	F	A				
									44B	S	G	I	S				
44B	A	1	C	3	N			1	44H	N	G	I	S	1	0	44A	
44C	A	1	C	3	T			1	44A	S	G	C	S	1	0	44H	
44H	B	1	C	3	T			1	44C	S	G	C	S	1	1	44K	44B
44K	C	4	C	3	T			1	44H	Y	G	F	S	4	0	23A	99F
																44D	
																26A	
44D	E	1	N	3	N			1	44K	Y	N	F	S	0	0	99F	70B
46A	A	4	B	3	T			4	23A	S	N	F	A	1	0	46C	
									46M	S	G	I	S				
									46P	S	G	I	S				
									46N	N	T	F	A				
46C	A	1	B	3	T			1	46A	S	G	C	S	1	0	46F	
46F	B	1	A	3	T			1	46C	S	G	C	S	1	1	46L	46G
46G	A	1	C	3	T			1	46F	N	G	I	S	1	0	46M	
46L	B	2	A	3	T			1	46F	S	G	F	S	2	1	46N	46P
																48B	
46M	E	4	N	3	T			1	46G	S	G	I	S	1	4	46A	46A
																46C	
																46F	
																46G	
46N	C	2	B	3	T			1	46L	Y	G	F	S	2	1	46R	46A
																42M	
46P	A	1	C	3	N			1	46L	N	G	I	S	1	0	46A	



Table B-1 (Continued)

----BOX-----   ---GENERAL DATA---   -----PREDECESSORS-----   ---SUCCESSORS---																	
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QWYS	QNN	YSBX	NBX
46R	A	2	C	3	N			2	46N	Y	N	F	A	1	0	25C	
46S	A	2	B	3	N			2	46T	N	N	I	S				
46T	B	5	A	3	N			1	25C	S	N	C	A	1	0	46T	
									42N	S	N	F	A				
									46S	S	N	C	S	5	1	48A	46R
																46W	
																28A	
																72A	
46W	E	1	N	3	N			1	46T	Y	N	F	A	0	0	99F	
48A	A	3	C	3	N			3	48D	Y	N	F	A	1	0	52A	
									46T	Y	N	F	A				
									52E	N	N	I	S				
48B	A	1	C	3	T			1	46L	Y	G	F	S	1	0	48D	
48D	C	1	B	3	T			1	48B	S	G	F	S	1	0	48A	99F
50A	A	2	C	1	N			2	42L	Y	N	F	A	1	0	50C	
									50E	N	N	I	S				
50C	A	1	A	1	N			1	50A	S	N	C	S	1	0	50E	
50E	B	1	A	1	N			1	50C	S	N	C	S	1	1	50H	50A
50H	A	1	C	1	N			1	50E	Y	N	F	S	1	0	54A	
52A	A	4	C	3	N			1	48A	S	N	C	A	4	0	52B	
																52C	
																52D	
52B	A	1	A	3	N			1	52A	S	N	C	A	1	0	52E	
52C	A	1	A	3	N			1	52A	S	N	C	A	1	0	52E	
52D	A	4	B	3	N			1	52A	S	N	C	A	4	0	52H	
																52L	
																52F	
																52M	

Table B-1 (Continued)

----BOX-----				---GENERAL DATA---				-----PREDECESSORS-----				----SUCCESSORS-----					
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QNYs	QNN	YSBX	NBX
52G	A	2	C	3	N			1	52A	S	N	C	A	1	0	52H	
52E	B	3	A	3	N			2	52J	N	N	I	S	3	1	52H 48A	
									52B	S	N	C	A			52M	
									52C	S	N	C	A			52F	
52F	E	2	N	3	N			2	52E	Y	N	F	A	0	0	99F	
52H	A	3	A	3	N			3	52G	S	N	F	A	1	0	52J	
									52E	Y	N	F	A				
									52D	S	N	F	A				
52J	B	1	A	3	N			1	52H	S	N	C	S	1	1	52P 52G	
52L	A	1	C	3	N			1	52D	S	N	F	S	1	0	52M	
52M	A	3	B	3	N			3	52E	Y	N	F	A	1	0	52P	
									52D	S	N	F	A				
									52L	S	N	F	A				
52P	A	2	A	3	N			2	52M	S	N	F	A	1	0	52Q	
									52J	Y	N	F	A				
52Q	A	1	A	3	N			1	52P	S	N	F	S	1	0	52R	
52R	A	2	B	3	N			2	52Q	S	N	F	S	1	0	52U	
									52S	S	N	I	S				
52S	A	1	C	3	N			1	52W	N	N	I	S	1	0	52R	
52U	A	1	C	3	N			1	52R	S	N	C	S	1	0	52V	
52V	A	1	A	3	N			1	52U	S	N	C	S	1	0	52W	
52W	B	2	A	3	N			1	52V	S	N	C	S	2	1	52X 52S	
																52Y	
52X	A	1	C	3	N			1	52W	Y	N	F	S	1	0	52Z	
52Y	A	1	B	3	N			1	52W	Y	N	F	S	1	0	52Z	
52Z	A	2	A	3	N			2	52Y	S	N	F	A	1	0	80F	
									52X	S	N	F	A				
53A	A	1	B	1	N			1	4A	S	N	F	A	1	0	54A	
53C	A	1	B	1	N			1	4A	S	N	F	A	1	0	54A	

Table B-1 (Continued)

----BOX-----				--GENERAL DATA---				-----PREDECESSORS-----				----SUCCESSORS-----					
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QNYs	QNN	YSBX	NBX
54A	A	5	C	1	N			5	50H	S	N	F	A	1	0	54E	
									53A	S	N	F	A				
									53C	S	N	F	A				
									54D	S	N	I	S				
									46T	Y	N	F	A				
54E	B	1	C	1	N			1	54A	S	N	C	S	1	1	54H	54D
54D	A	1	C	3	N			1	54E	N	N	I	S	1	0	54A	
54H	A	2	B	1	N			2	54E	Y	N	F	A	1	0	54K	
									54L	S	N	I	S				
54K	B	1	A	1	N			1	54H	S	N	C	S	1	1	54M	54L
54L	A	1	C	3	N			1	54K	N	N	I	S	1	0	54H	
54M	B	1	A	1	N			1	54K	Y	N	F	S	1	1	54N	54T
54N	B	1	A	3	N			1	54M	Y	N	F	S	1	1	54P	54T
54P	A	1	B	3	N			1	54N	Y	N	F	S	1	0	54Q	
54Q	A	2	B	3	N			2	54P	S	N	F	S	1	0	54S	
									54S	N	N	I	S				
54S	B	1	A	1	N			1	54Q	S	N	C	S	1	1	54T	54Q
54T	A	4	C	1	N			4	54M	N	N	F	R	1	0	54V	
									54N	N	N	F	R				
									54S	Y	N	F	R				
									54V	N	N	I	S				
54V	B	1	A	1	N			1	54T	S	N	C	S	0	1	99F	54T
60A	H	3	C	1	N			1	4S	S	N	F	S	3	0	60B	
																16C	
60B	H	4	C	1	N			1	60A	S	N	F	S	4	0	62A	
																80A	
																80C	
																80E	
																80D	

Table B-1 (Continued)

----BOX-----			--GENERAL DATA----			Sub	-----PREDECESSORS-----			----SUCCESSORS-----						
ID	Type	Lns	Doer	Lvl	Grp	Net	Quan	Box	Exit	Grp	PM	Strt	QNYs	QNN	YBX	NBX
62A	H	3	B	1	N		2	4S 60A	S S	N N	F F	A A	3	0	62B 18J 44A	
62B	H	4	C	1	N		1	62A	S	N	F	S	4	0	80A 80C 80E 80D	
62C	H	1	C	1	D		1	18F	Y	G	I	S	1	0	18J	
66B	A	2	B	1	N		2	4L 66F	Y N	N N	F I	S S	1	0	66D	
66D	A	1	B	1	N		1	66B	S	N	C	S	1	0	66F	66B
66F	B	1	B	1	N		1	66D	S	N	C	S	0	1	99F	
70A	A	1	C	3	N		1	12H	Y	N	F	S	1	0	70B	
70B	A	3	C	3	N		3	70A 44K	S Y	N N	F F	A A	1	0	70C	
70C	A	1	A	3	N		1	70E	N	N	I	S	1	0	70E	70B
70E	B	1	A	3	N		1	70B 70C	S S	N N	C C	S S	1	1	72C	
72A	A	2	C	3	N		2	46T	Y	N	F	A	1	0		
72C	A	2	C	1	N		1	72C 70E	S Y	N N	F F	A S	2	0	74A 72A 74A	
74A	A	3	B	1	N		3	72C 72A	S S	N N	F F	A A	1	0	74C	
74B	A	1	C	1	N		1	74B	S	N	I	S	1	0	74A	74B
74C	B	1	A	1	N		1	74C 74A	N S	N N	C C	S S	1	1	82E	

Table B-1 (Continued)

----BOX-----				--GENERAL DATA--				-----PREDECESSORS-----				----SUCCESSORS----					
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QNYS	QNN	YSBX	NBX
80A	A	3	A	3	N			3	62B	S	N	F	A	1	0	80G	
									60B	S	N	F	A				
80C	A	3	A	1	N			3	26A	S	N	F	A	1	0	80H	
									62B	S	N	F	A				
									60B	S	N	F	A				
80D	E	3	N	3	N			3	26A	S	N	F	A	0	1	99F	66F*
									62B	S	N	F	A				
									60B	S	N	F	A				
80E	A	4	C	3	N			4	26A	S	N	F	A	1	0	82B	
									62B	S	N	F	A				
									60B	S	N	F	A				
									26A	S	N	F	A				
80F	E	2	N	3	N			2	82C	N	N	I	S				
									80J	Y	N	F	A	0	0	99F	
80G	A	1	A	3	N			1	52Z	S	N	F	A				
80H	A	1	A	3	N			1	80A	S	N	F	S	1	0	80J	
80J	B	3	A	3	N			3	80C	S	N	F	S	1	0	80J	
									80G	S	N	F	A	3	1	80F	80L
									80H	S	N	F	A			82A	
80L	A	1	A	3	N				80P	S	N	I	S			82B	
80N	A	1	C	3	N			1	80J	N	N	I	S	1	0	80N	
80P	A	1	A	3	N			1	80L	S	N	I	S	1	0	80P	
									80N	S	N	I	S	1	0	80J	
82A	A	2	C	3	N			2	80J	Y	N	F	A	1	0	82G	
									52Z	S	N	F	A				
82B	A	3	A	3	N			3	80J	Y	N	F	A	1	0	82C	
									52Z	S	N	F	A				
82C	B	4	A	3	N			1	80E	S	N	C	A	4	1	82E	80E
									82B	S	N	C	S			82J	
																82K	
																82L	

Table B-1 (Concluded)

----BOX-----   --GENERAL DATA---   -----PREDECESSORS-----   ---SUCCESSORS---																	
ID	Type	Lns	Doer	Lvl	Grp	Net	Sub	Quan	Box	Exit	Grp	PM	Strt	QNYS	QNN	YSBX	NBX
82E	A	3	A	3	N			3	82C	Y	N	F	A	1	0	82G	
									28A	S	N	F	A				
									74C	Y	N	F	A				
82G	A	2	B	3	N			2	82A	S	N	F	A	1	0	82Q	
									82E	S	N	F	A				
82J	A	1	C	3	N			1	82C	Y	N	F	S	1	0	82P	
82K	A	1	C	3	N			1	82C	Y	N	F	S	1	0	82P	
82L	A	1	C	3	N			1	82C	Y	N	F	S	1	0	82P	
82P	A	3	B	3	N			3	82J	S	N	F	A	1	0	82Q	
									82K	S	N	F	A				
									82L	S	N	F	A				
82Q	A	3	C	3	N			3	82P	S	N	C	A	1	0	82R	
									82G	S	N	F	A				
									82U	S	N	I	S				
82R	A	1	C	3	N			1	82Q	S	N	C	S	1	0	82S	
82S	A	1	A	3	N			1	82R	S	N	C	S	1	0	82T	
82T	B	1	A	3	N			1	82S	S	N	C	S	1	1	82V	82U
82U	A	1	B	3	N			1	82T	N	N	I	S	1	0	82Q	
82V	A	1	A	3	N			1	82T	Y	N	F	S	1	0	82W	
82W	A	2	C	3	N			2	82V	S	N	F	S	1	0	82X	
									82Y	N	N	I	S				
82X	A	1	A	3	N			1	82W	S	N	C	S	1	0	82Y	
82Y	B	3	A	3	N			1	82X	S	N	C	S	0	1	99F	82W

\* Access via Special Event Box, only.

Table B-2 Software Acquisition Process Model  
Activity Box Parameter Data

ID	TYPE	GRP	SYS	DSGN	PRCH	TEST	SPRT	MANPOWER			SPRT	GOVERNMENT			Iterate Factor	DURATIONS		WAIT DAYS	NOTE
								CONTRACTOR	DEV	USR		DEV	USR	SPRT		DAYS	IT		
2A	A	M														10			1
4A	A	M	2	3	3	2	1									10			2
4C	A	M	2	2	2	2	2								5 5 5	12	3	1	3, 4
4E	A	M							6						5 5 5	10	3	1	5
4J	A	M	3	2	5	1	1		10	2	1				5 3 3	4	3	2	10
4M	A	M	2	1		1	1								5 5 5	3	6	3	8
4S	A	M	1	2	5	2	1								5 5 5	15			5
6A	A	M	2	5											10 5 3	31	3	2	10
6D	A	M	3	12	5	3									5 3 1	6	5	2	10
6F	A	M	3	3											8 5 5	8	5	3	11
6G	A	D	3	15		5									3 2 1	21	4	4	12
6H	A	D	3	15	2										5 5 5	9	4	2	12
6I	A	D	1	5					1						5 2 2	6	3	3	14
8A	A	D														9	5	3	14
10A	A	D	3	24											6 4 2	42	3	2	
10C	A	D	3	24	2										5 3 2	9	6	4	
10E	A	D	5	24	2	5	3								6 4 2	5	6	4	
10L	A	D	4	12												10	6	4	
12A	A	D	5	10	4	2			15	4	2					18	3	1	14
12J	A	D	3	10	6	1										18	5	2	14
14A	A	M	3	3	5										5 5 5	20	4	2	
14C	A	M	3	3	1	8	3								5 3 1	30	5	3	
16A	A	D		24	2	48									3 3 3	27			
16C	A	D		24	2	48	3								3 3 3	15	5	5	
18C	A	D		24	3	48	3								3 3 3	42	5	5	
18E	A	D		24	4	48	5								5 5 5	54	5	5	

Table B-2 (Continued)

---BOX---			---MANPOWER---										---GOVERNMENT---			---DURATIONS---			WAIT	NOTE
ID	TYPE	GRP	SYS	DSCN	PRGM	CONTRACTOR	TEST	SPRT	DEV	USR	SPRT	Iterate Factor	DAYS	IT	FCTR	DAYS				
18G	A	D						4				101010	15	5	5	5				
18J	A	D					8	2				10 6 4	15	6	4	2				
18H	A	D											12	10	10	10				
18N	A	D	1	1	1		1	1	15				12	10	10	10				
18Q	A	D											6	1	1	1				
20A	A	N											5	5	3	1				
20C	A	D											30	5	3	1				
22A	A	N						4					4	10	10	10				
23A	A	N											4	1	1	1				
24A	A	D											18	5	3	1				
24C	A	D											27	5	3	1				
26A	A	N											40	5	3	1				
28A	A	N											10	5	3	1				
40A	A	N											25					5		
40C	A	N											15					5		
40G	A	N											6							
40H	A	N											5							
40J	A	N											5							
42A	A	T											50							
42C	A	T											25					5		
42H	A	T											10							
42J	A	T											10							
42K	A	T											6							
42H	A	N											5							
42N	A	N											8							



Table B-2 (Continued)

BOX		GRP		SYS		DSGN		CONTRACTOR		MANPOWER		SPRT		DEV		USR		SPRT		Iterate		DAYS		IT FCTR		WAIT		NOTE	
ID	TYPE																												
44A	A		T																			8	7	5	3				
44R	A		T																			21	8	6	4				
44C	A		T																										
46A	A		T																			10	8	6	4				
46C	A		T																			16	8	6	4				
46G	A		T																			14	8	6	4				
46P	A		T																			6	8	6	4				
46R	A		N																			10	5	3	1				
46S	A		N																			6	5	3	1				
48A	A		N																			6	5	3	1				
48B	A		T																			16	1	1	1				
50A	A		N																										
50C	A		N																										
50H	A		N																										
52A	A		N																										
52B	A		N																			6	6	4	2		5		
52C	A		N																			5							
52D	A		N																										
52G	A		N																										
52H	A		N																										
52L	A		N																										
52M	A		N																			1							
52P	A		N																			6							
52Q	A		N																			1							
52R	A		N																			1							
52S	A		N																										
52U	A		N																			1							
52V	A		N																			1							

Table B-2 (Continued)

-----BOX-----			-----CONTRACTOR-----			-----MANPOWER-----			-----GOVERNMENT-----			-----DURATIONS-----			WAIT DAYS	NOTE
ID	TYPE	GRP	SYS	DSGN	PRGM	TEST	SPRT	DEV	USR	SPRT	Iterate Factor	DAYS	IT	FCTR		
52X	A	N										3				
52Y	A	N										2				
52Z	A	N										1				
53A	A	N														
53C	A	N														
54A	A	N														
54D	A	N														
54H	A	N														
54L	A	N														
54P	A	N														
54Q	A	N														
54T	A	N														
60A	H	N	1 5	1	4 5		4					40			6	
60B	H	N	1 2	1 5	2		4 5					500				
62A	H	N	1 5	1	4 5	2 5	4					40			7	
62B	H	N	1 2	1 2	1 1	1 5	4 5					400				
62C	H	D			1	3						12	5 3	2		
66B	A	N	2 2	1 5		2 5	1	5 4	1	1		5			30	
66D	A	N	3 7	3 3			1	8	2	1		4			9	
70A	A	N														
70B	A	N														
70C	A	N														
72A	A	N														
72C	A	N														
74A	A	N														

Table B-2 (Continued)

ID	BOX TYPE	GRP	MANPOWER										WAIT DAYS	NOTE
			DSGN	PRGM	CONTRACTOR	TEST	SPRT	DEV	USR	SPRT	Iterate Factor	IT FCTR		
74B	A	N												
80A	A	N												
80C	A	N												
80E	A	N												
80G	A	N												
80H	A	N												
80L	A	N												
80N	A	N												
80P	A	N												
82A	A	N												
82B	A	N												
82E	A	N												
82G	A	N												
82J	A	N												
82K	A	N												
82L	A	N												
82P	A	N												
82Q	A	N												
82R	A	N												
82S	A	N												
82U	A	N												
82V	A	N												
82W	A	N												
82X	A	N												

Table B-2 (Concluded)

ID	BOX		GRP	MANPOWER				CONTRACTOR			GOVERNMENT			iterate		DURATIONS		WAIT		NOTE
	TYPE			SYS	DSGN	PRGM	TEST	SPRT	DEV	USR	SPRT	Factor	DAYS	IT	FCTR	DAYS				
96A	A		N																	
96B	A		N																	
96C	A		N																	
96G	A		N																	
96H	A		N																	
96K	A		N																	
96P	A		N																	
96S	A		N																	
96V	A		N																	
96Z	A		N																	

Table B-3

Software Acquisition Process Model  
Decision Box Parameter Data

-----BOX-----			-----YES EXIT PROB-----				WAIT	NOTES
ID	TYPE	GROUP						
4G	B	N	80	100	100	100		8
4L	B	N	80	90	90	90		8
6E	B	N	20	60	80	100		
6I	B	D	70	90	100	100		12
6P	B	N	50	0	0	0		13
8C	B	D	90	100	100	100		14
10E	B	D	20	50	80	95		
10J	B	D	20	50	80	95		
12C	B	D	80	90	90	95		
12G	B	D	90	90	90	100		
18F	B	D	5	0	0	0		
18I	B	D	100	80	40	20		
18K	B	D	80	60	40	25		
18L	B	D	15	0	0	0		
20E	B	N	70	90	95	100		
24E	B	D	70	90	95	100		
42E	B	T	40	60	75	90		
44H	B	T	5	25	50	75		
46F	B	T	90	100	100	100		
46L	B	T	20	40	60	90		
46T	B	N	80	90	100	95		
50E	B	N	50	70	90	100		
52E	B	N	80	90	100	100		
52J	B	N	75	95	100	100		
52W	B	N	90	100	100	100		
54E	B	N	10	25	50	75		
54K	B	N	20	40	75	90		
54M	B	N	75	00	00	00		
54N	B	N	80	00	00	00		
54S	B	N	60	90	95	100		

Table B-3 (Concluded)

-----BOX-----			-----YES EXIT PROB-----				WAIT	NOTES
ID	TYPE	GROUP						
54V	B	N	40	70	95	100		
66F	B	N	0	0	0	0		9
70E	B	N	50	70	90	100		
74C	B	N	40	60	70	95		
80J	B	N	65	85	95	100		
82C	B	N	75	90	100	100		
82T	B	N	75	80	90	100		
82Y	B	N	75	90	95	100		
96E	B	N	60	80	95	100		
96F	B	N	75	75	75	75		
96J	B	N	65	85	95	100		
96L	B	N	60	60	60	60		
96N	B	N	75	50	25	25		
96Q	B	N	50	50	50	50		
96T	B	N	65	85	90	90		
96W	B	N	75	60	50	25		

Table B-4

Software Acquisition Process Model  
Counter & Special Event Box Parameter Data

-----BOX-----			FUNCT	EVENT LABEL	PARAMETER	NOTES
ID	TYPE	GRP				
6L	E	D	M	PDR (DTN.NO)		
8E	C	D				15
10F	C	D				15
12E	C	D				15
12F	E	D	M	CDR (DTN.NO)		
12H	C	D				15
18H	C	D				15
18P	C	D				15
18R	E	D	M	CCI&C (DTN.NO)		
42B	C	T				15
42L	C	T				15
44D	E	N	M	DRY RUN - END		
44K	C	T				15
46M	E	T	R			
46N	C	T				15
46W	E	N	M	FQT - END		
48D	C	T				15
52F	E	N	M	FCA - START		
80D	E	N	P		pY=100	
80F	E	N	M	PCA - START		

## APPENDIX C

### RATIONALE FOR PROCESS MODEL DEVELOPMENT

This appendix repeats attachment 2 of MITRE letter to ESD/TOIT D75-173, "Proposed FY 79 Project 5720 Tasks in Software Resource and Cost Prediction", 17 July 1978, because the letter is unavailable to most readers of this report. The attachment, entitled "Advantages of Software Acquisition Process Model Development," is as follows.

FY 79 MITRE work to date strongly indicates substantial potential benefits from developing and applying in selected ESD-managed acquisition programs a simulation model of the software acquisition process. The type of software acquisition process model considered would represent explicitly (e.g., in flowchart form) the different Program Office (PO) and contractor activities that ESD-managed software acquisition entails, and how these different activities interact. It would represent activity sequences, repetitions of such sequences, alternatives, concurrency, and delays (e.g., due to waits for essential inputs). In this respect the model would somewhat resemble PERT, but without PERT's restrictions on loop representation, etc.

Besides this process logic, the model would also accept, wherever appropriate, a definite parameter value or distribution of values for each activity's resource requirements (e.g., manpower, computer time), elapsed time, and dollar cost. The software acquisition process model would also accept parameter values representing the probabilities of transition among the different activities; for example, the chances of 1, 2, or more document review cycles could be represented. (An initial set of these parameter values would be included so users would need to change only the values pertinent to each use).

Early in an acquisition program's life cycle a flowchart of the acquisition program's planned software-related activities would be developed, and coded in a computer simulation language. The parameter values associated with the program's activities and transitions among activities would also be estimated and put into the model. These flowcharts and local estimates would be prepared for, and reviewed by, the Program Manager (PM), based on current acquisition program plans. The model thus defined would then undergo computer simulation to develop initial predictions of the acquisition program's overall software-related requirements, schedules, and dollar costs, in total, by phase, and by function. In addition, the simulation would summarize delays, by function, and other indicators of potential problems. The software acquisition process model logic could be modified, the parameter values altered, and the simulation program rerun, to explore the effects of alternative policies and the overall estimates' sensitivities to particular local estimates.

Such early use of a software acquisition process model would force early definition of concrete plans, and early development of specific local



estimates. This alone would be a major benefit of using a software acquisition process model in an acquisition program. While the process model concept mandates no specific level of detail, and in fact allows that level to be varied at will, the effort necessary to develop, and to review at any reasonable level, acquisition program logic, local estimates, and simulation results, would be sure to expose some vagueness, ambiguity, and oversight, which if then corrected would prevent serious later problems.

In addition, software acquisition process model simulation results would generally be more accurate than estimates derived by more general methods (e.g., using typical parametric cost estimation models), because the former would reflect the specific acquisition program's plans and policies. Where desired, the simulation program could also develop envelopes of values around these results, as measures of their uncertainty.

Perhaps most important, the simulation results would be credible, because they would be based on flowcharts and local estimates understood and approved by PO management. The software acquisition process model logic could normally be verified by flowchart inspection. The local estimates would in some cases be verifiable immediately as relatively simple facts, or as assumptions; in other cases verification would require investigation, but usually would be relatively straightforward because it would involve a local variable. In contrast, typical parametric models available today are mysterious because they represent no well-defined mechanisms, and unverifiable (at desirable levels of accuracy) because appropriate data for their rigorous statistical testing is unavailable. Only a systematic data collection effort (like the planned ESD Software Data Reporting System), applied over many years, can substantially mitigate this problem.

As an acquisition program evolves, actual resource expenditures, elapsed times, and dollars spent become known, and the actual (vs. planned) logic of past activities becomes clear. Based on such experience a PO changes its acquisition program plans, and updates its predictions of future activities' costs. In an acquisition program application, the software acquisition process model would be revised at convenient intervals to reflect past event logic, actual expenditures, actual elapsed times, revised plans, and new estimates based on all of the above. After each such set of changes, the revised acquisition process model would provide the PO with a series of overall predictions, generally of increasing accuracy. Each of these would be based on the best experience then available, and on then current plans.

Like the PO's original version of the software acquisition process model, any subsequent version could be altered and simulated to explore the effects of proposed policy changes and different local estimates. In this way, the model could become a powerful aid to source selection, provided the acquisition program's RFP(s) directed the offerors to provide the necessary input in their proposals, and that the source selection schedule allowed enough time for the necessary model modifications, parameterization, and simulation runs. To help in source selection, the PO's current version of the software acquisition process model would be modified and simulated (where possible) to reflect each offeror's plans for software-related work, and parameterized to reflect his resource, elapsed time, and dollar cost

estimates. These activities should reveal omissions, inconsistencies, and unexpected implications that current proposal review methods rarely catch. The model would also be modified, parameterized, and simulated to reflect the Government's best assessment of the activities, resources, elapsed times, and dollar costs, necessary to do the RFP-specified work, based on the Government's understanding of each offeror's proposed approach, modified for completeness, consistency, and realism. The results of this modification and simulation would considerably improve the realism of contract negotiations by sensitizing the Government negotiators to proposal flaws now often overlooked. For similar reasons, Source Selection Board recommendations would be improved.

After source selection, the PO's then standard model version would be altered to reflect the selected contractor's development approach. It would subsequently be updated, as indicated earlier, to reflect actual contractor performance, and used to predict the effects of proposed changes in contract scope, contractor organization, and contractor resource allocation. For example, a contractor's justification of his estimates to implement a proposed contract change (e.g., an ECP) could be subject to verification by PO software acquisition process model use. As a result, excessive ECP implementation cost estimates, now a common problem, would often be avoided.

Besides these specific acquisition program applications, a software acquisition process model could greatly aid investigation of the quantitative effects on resource requirements, elapsed times and dollar costs of varying the program development or maintenance environment (e.g., by the use of structured programming, improved compilers, or other development tools), of imposed schedules, of computer program type and complexity, and of many other factors. In this type of use a software acquisition process model would be altered or parameterized to reflect each factor or appropriate combination of factors. The model would then be simulated, the effects noted, and the causes investigated. Multiple runs using different estimates could establish the sensitivity of the overall process, or of important subprocesses (e.g., detailed design, coding, test and integration), to different assumed values of the factor(s) under investigation. In effect, this kind of model application would provide a controlled experimental environment not achievable in practice. It could be supplemented by limited actual experiments to obtain realistic parameter values.

Probably the most important near term use of this kind would apply the software acquisition process model to help revise and redefine, in FY 79 and later, the prototype set of software-related cost reporting elements defined under a FY 78 MITRE task. One basic objective of this work is to define a minimum number of elements that nevertheless capture and distinguish all important kinds of software-related resource expenditures, elapsed times, and costs. The prototype element set definitions are being established in large part as a result of a few persons' software development and software acquisition management experience. While widespread critical review will doubtless suggest many improvements, and while planned pilot application will identify others, these activities could nevertheless overlook important deficiencies.

Software acquisition process modeling could significantly help the element redefinition process. As flowcharts of the principal kinds of software-related activity are developed, and as estimates are prepared of the individual activities and their transition probabilities, it often becomes clear what activities should be grouped together, and what their approximate relative magnitudes are. This has already occurred in some cases during our preparation of sample software acquisition process flowcharts during the FY 78 model feasibility study. Definition of other groups of processes is expected to yield further insights of this kind. Actual model simulation would be even more effective. Since the software cost reporting elements should seldom be changed once the Software Data Reporting System is widely applied, these elements should be well defined and reasonably stable before then. In the absence of a respectable data base (which the Software Cost Reporting System is designed to collect), the software acquisition process model seems a very promising aid to judgment and limited experience in defining the elements effectively.

APPENDIX D  
PARAMETER VALUE  
CALIBRATION

This Appendix describes the numerical methods used to calibrate the activity duration data, and provides the detailed results obtained. Figure D-1, Activity Duration Calibration, shows the calibration procedures applied to each unit sequence of the mainstream developmental activity for the contract period up through Formal Qualification Testing (FQT). Figure D-2, Activity Duration Calibration Summary, shows the sequential relationship between the unit sequences and the consequent overall timing. Each of the figures is further described below.

As shown in these figures, four different activity times were used or developed. These are:

- a. Initial value - the estimated elapsed time initially established.
- b. Initial iterated value - the initial estimated time, which has been lengthened to include the effect of iteration.
- c. Calibrated value - the time assigned as a result of the calibration procedure.
- d. Calibrated iterated value - the calibrated time with the lengthening effect of iteration added.

In Figure D-1 the overall process was divided into a contiguous set of unit sequences. Boundaries between the sequences were selected to avoid (or minimize) the interruption of any iteration loops. Each sequence was then timed by following its flow paths in conformance with the Decision Box probability (pY) data. This was done by assuming a sample of 100 runs during which a count was kept of how many times each Activity Box (A-box) was entered per each iteration count. This permitted the total time spent in each A-box to be calculated. This is shown (per iteration) on the diagram. By dividing the total time by the number of runs (i.e., 100), the "iterated" time for each A-box was determined. Thus the most likely time to traverse each unit sequence became the sum of the iterated times for the A-boxes which form the unit sequence.

Figure D-2 shows the overall flow. Sheet 1 represents the workdays for each sequence based on calibrated activity times (without iteration). The number in parentheses represents the cumulative work days (based on the longest path) for each sequence. Sheet 2 depicts the same kind of information using the initial iterated times. The same information, but based on calibrated iterated time, is shown in Figure 10 of the report proper.

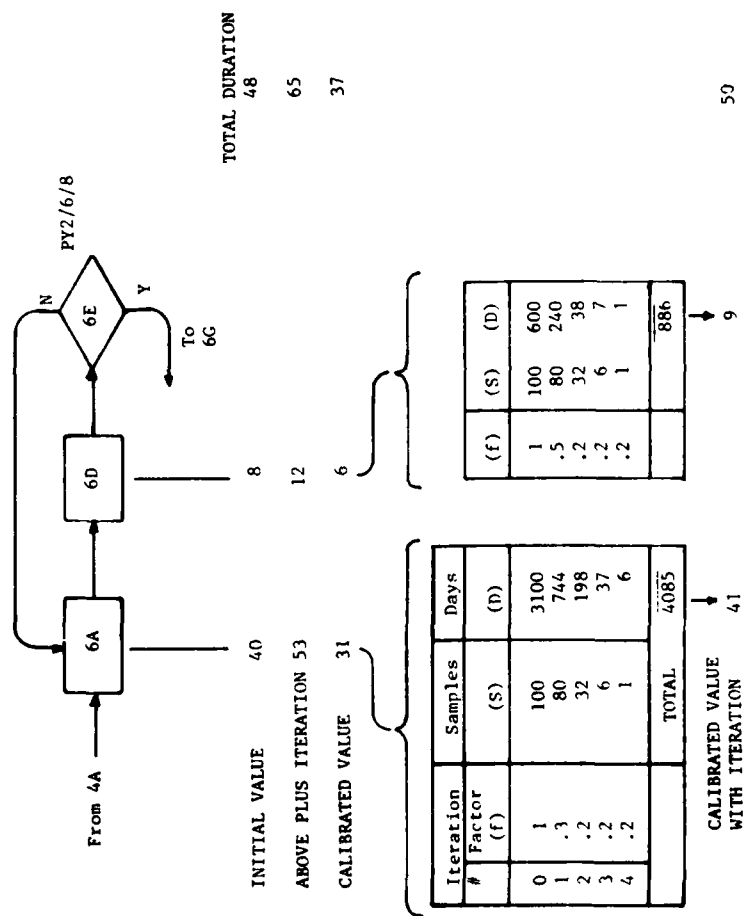


Figure D-1. Activity Duration Calibration  
Sheet 1



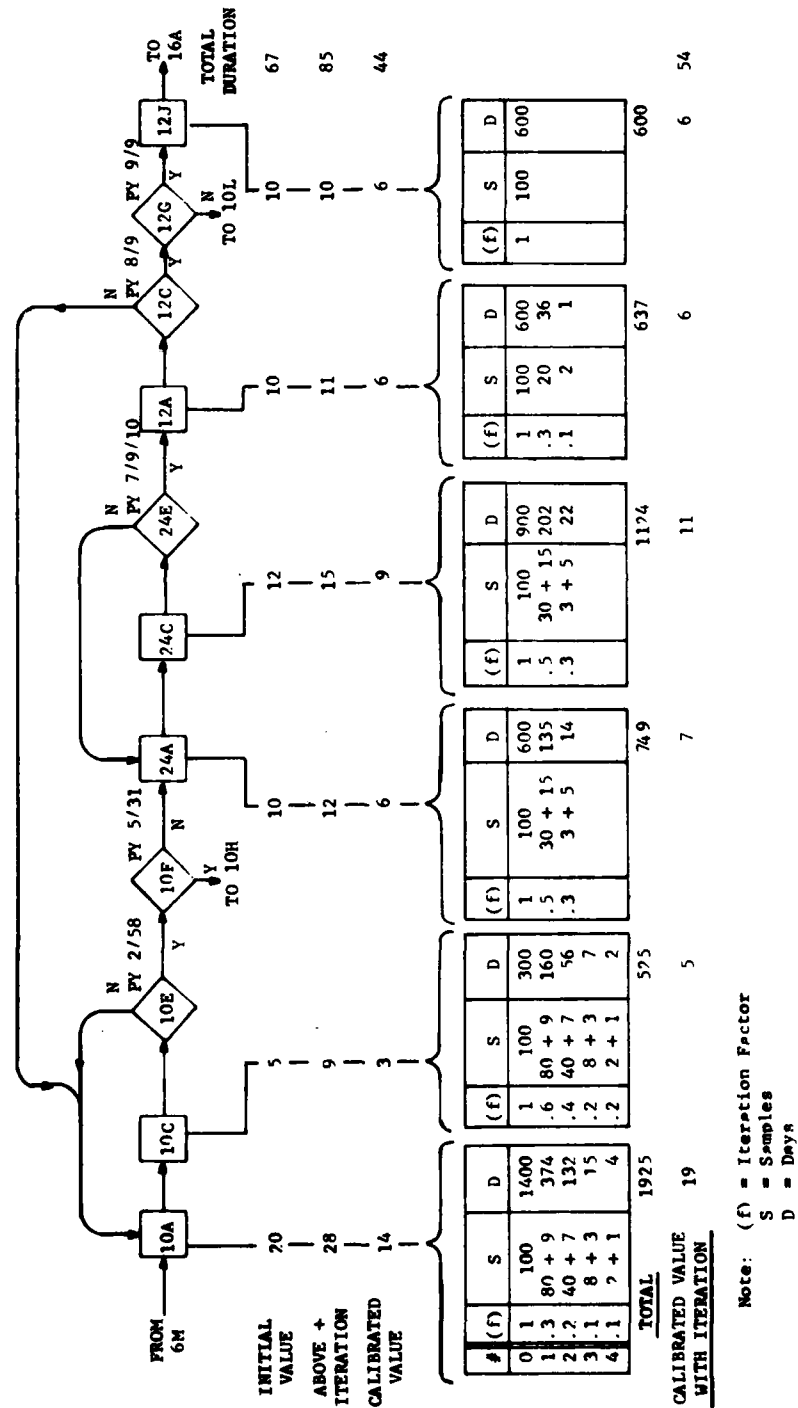


Figure D-1. Activity Duration Calibration  
Sheet 3

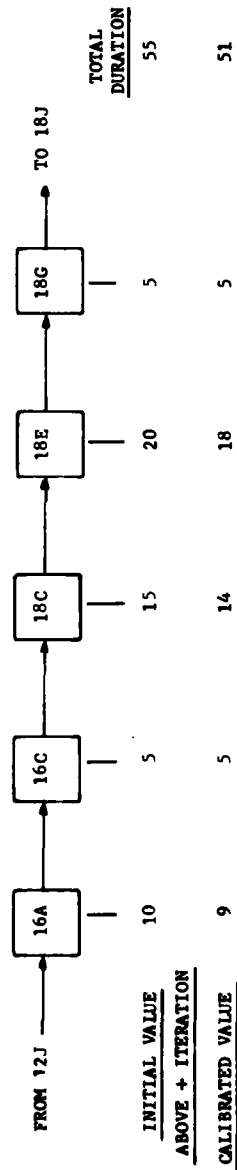
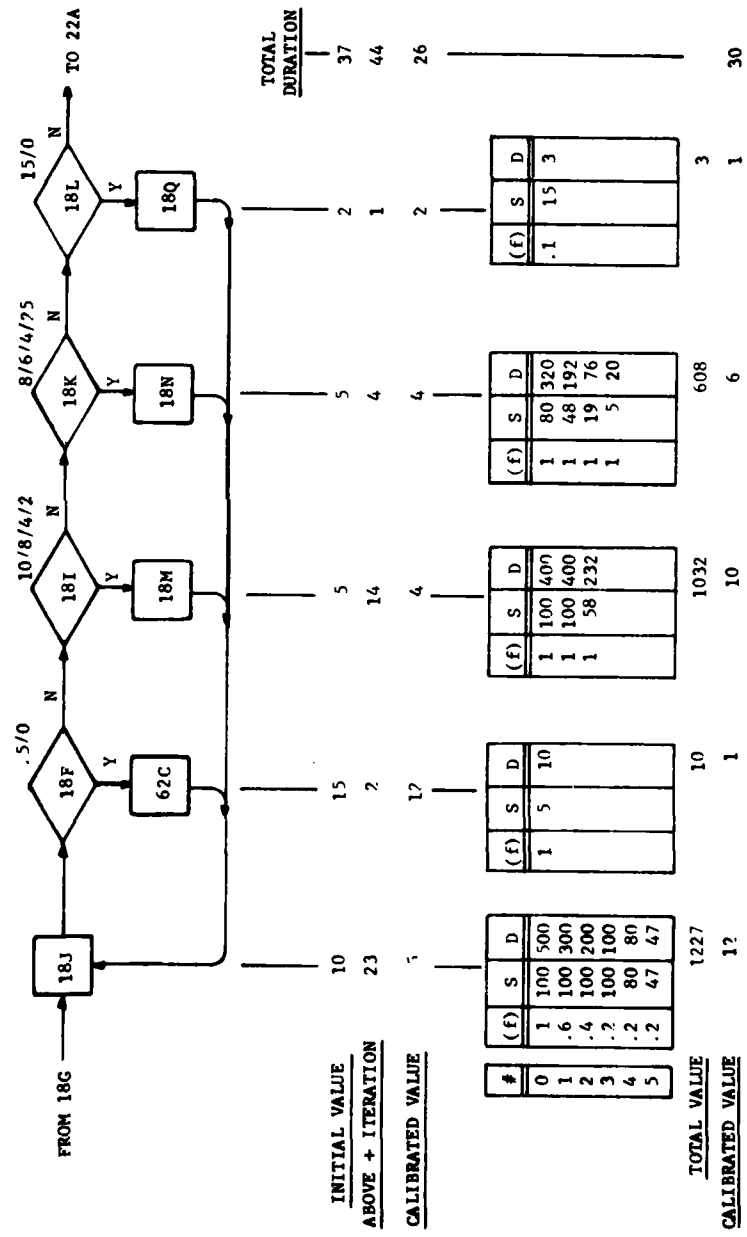


Figure D-1. Activity Duration Calibration  
Sheet 4

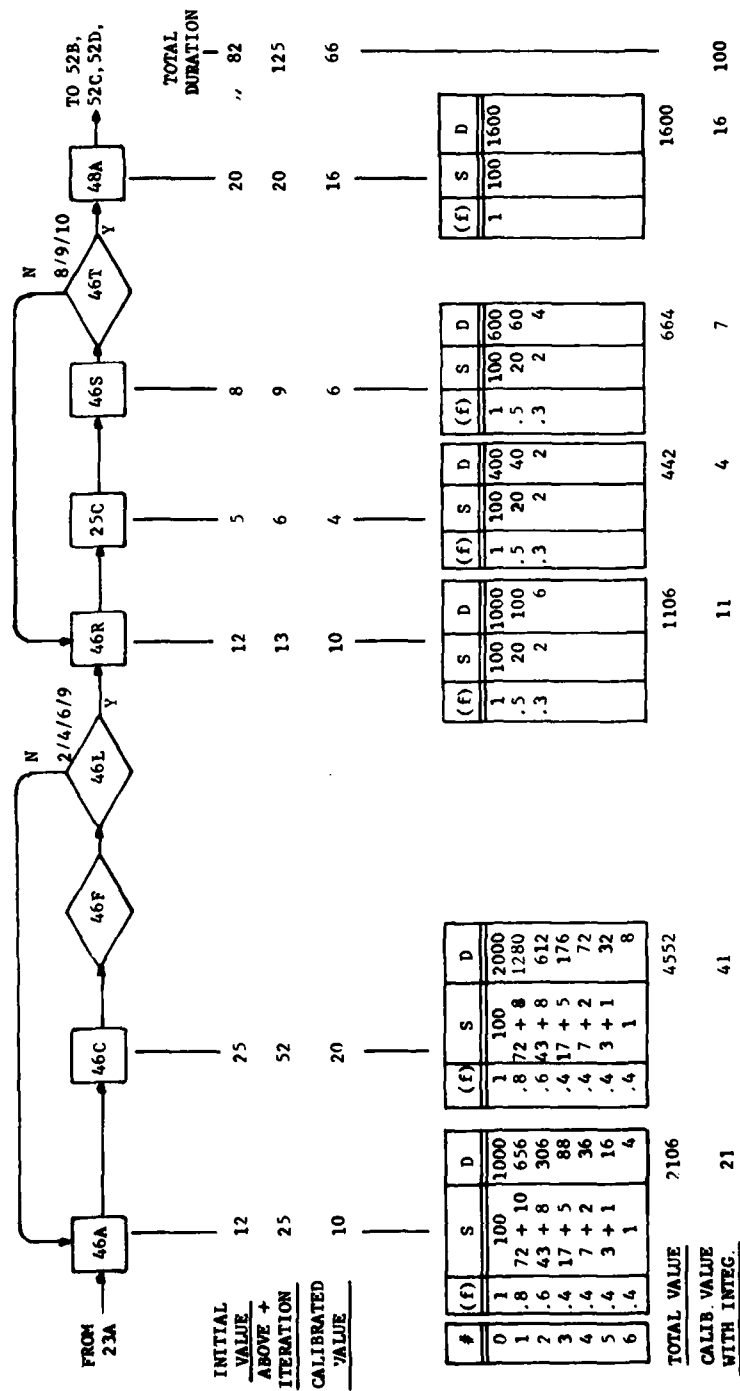




Note : (f) = Iteration Factor  
 S = Samples  
 D = Days

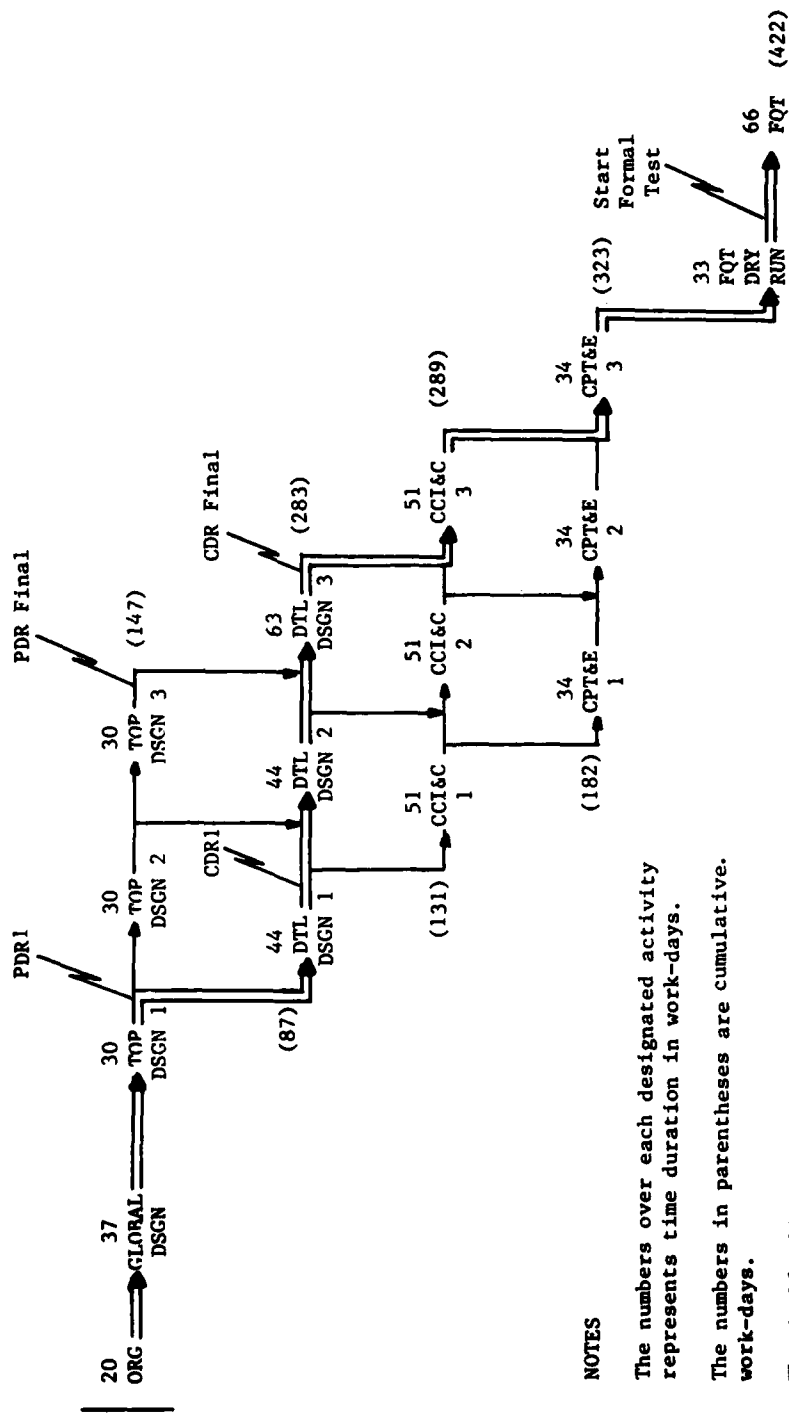
Figure D-1. Activity Duration Calibration  
 Sheet 5





Note : (f) = Iteration Factor  
 S = Samples  
 D = Days

Figure D-1. Activity Duration Calibration  
 Sheet 7



#### NOTES

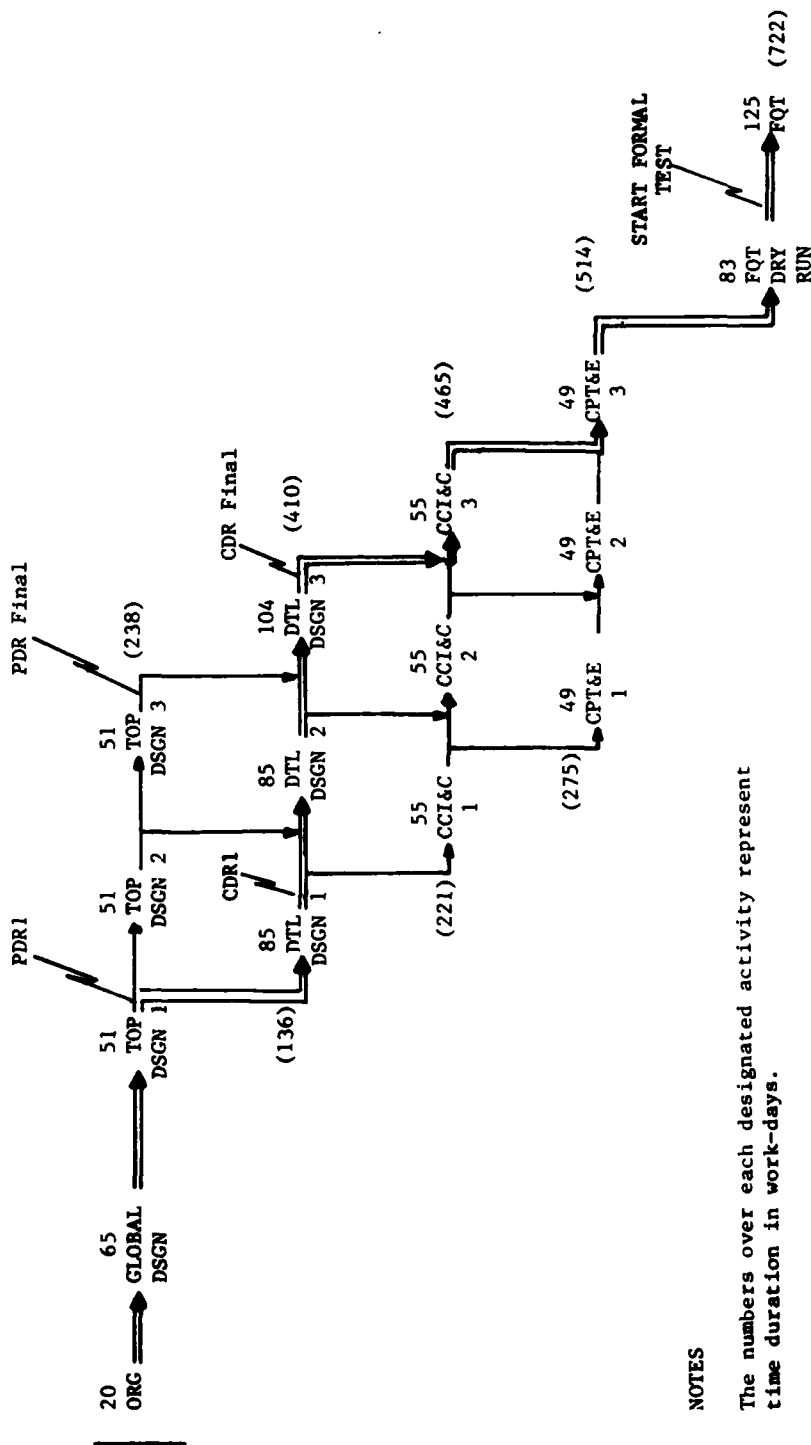
The numbers over each designated activity represents time duration in work-days.

The numbers in parentheses are cumulative work-days.

The double line represents the main path.

Timing values shown here are based on the calibrated activity durations (no iteration) as shown on Figure D-1.

Figure D-2. Activity Duration Calibration Summary  
Sheet 1 - Calibrated Activity Durations, With No Iteration



#### NOTES

The numbers over each designated activity represent time duration in work-days.

The numbers in parentheses are cumulative work-days.

The double line represents the main path.

Timing values shown here are based on calibrated activity durations (with iteration).

Figure D-2. Activity Duration Calibration Summary  
Sheet 2 - Initial Estimated Activity Durations, With Iteration

APPENDIX E  
SIMULATION PROGRAM LISTING EXTRACTS

This appendix provides extracts from the compilation listing of the simulation computer program thus far completed.

The computer printouts provided herein comprise the following three portions:

- a. The portions of the Global Data Base prepared to date are defined within the program Preamble. This appears on pages 1-6 of the listing; these page numbers appear on the top right corner of each page.
- b. The Data Input Processor routines prepared to date are included on listing pages 11-34. Brief descriptions and definitions of local variables are provided within each routine. Most of the cross reference listings that normally follow each routine were excised to reduce the size of this appendix.
- c. The remainder of the appendix shows the output obtained from a test run of the program. Much of this output is for data checking, and will not be included in the final program version.

LINE CACI SIMSCRIPT II.S RELEASE 8H 11/09/79 PAGE 1

```

1  **
2  **
3  ** SOFTWARE ACQUISITION PROCESS MODEL SIMULATION PROGRAM MODEL 0
4  ** VERSION DATED 9 NOVEMBER 1979, REVISION 3
5  ** H. MCLEANE
6  **
7  ** SOFTWARE ACQUISITION PROCESS MODEL SIMULATION PROGRAM MODEL 0
8  **
9  PREAMBLE
10 **
11 ** NORMALLY, MODE IS INTEGER
12 **
13 ** THE SOURCE OF EACH ATTRIBUTE IS SHOWN BY A CODE IN THE FIRST COLUMN OF THE
14 ** FIRST OR ONLY COMMENT ASSOCIATED WITH THAT ATTRIBUTE, AS FOLLOWS:
15 ** 'I' - INPUT OR INPUT-DERIVED
16 ** 'C' - COMPILER-GENERATED
17 ** 'S' - DEVELOPED DURING EACH SIMULATION PASS
18 ** 'B' - DEVELOPED AFTER EACH SIMULATION REPETITION
19 **
20 ** EVENT NOTICES INCLUDE END, REPETITION AND END, SIMULATION
21 **
22 ** EVERY FLOW HAS
23 ** S FOR THE OPERAND BOX TO WHICH THIS
24 ** EVENT NOTICE APPLIES:
25 ** S DTH.NO OF OPERAND BOX
26 ** S PHA OF OPERAND BOX: 0 - FORWARD; 1 - ITERATIVE
27 ** S 0 - DECISION "YES"; 1 - DECISION "NO"
28 ** S POINTER TO OPERAND BOX
29 ** IN WORD 5
30 ** EVERY BOX, PROC HAS
31 ** S FOR THE OPERAND BOX TO WHICH THIS
32 ** EVENT NOTICE APPLIES:
33 ** S DTH.NO OF OPERAND BOX
34 ** S PHA OF OPERAND BOX: 0 - FORWARD; 1 - ITERATIVE
35 ** S POINTER TO OPERAND BOX
36 ** IN WORD 5
37 ** PRIORITY ORDER IS FLOW, BOX, PROC, END, REPETITION AND END, SIMULATION
38 PERMANENT ENTITIES
39 **
40 ** EVERY BOX HAS
41 ** A BOX.ID
42 ** IN ARRAY 1,
43 ** A (BOX.TYPE(1-N),
44 **
45 **
46 **
47 ** SUBNETWORK (5-8),
48 **
49 ** LEVEL (9-12),
50 ** DTH (13-16),
51 **
52 ** DOER (15-16),
53 **

```

\*\*I 4-CHARACTER BOX IDENTIFIER  
 \*\*I 1 - THE PSD PHASE ORIGIN;  
 \*\* 2 - ACTIVITY; 3 - HELPER; 4 - SUBPROCESS;  
 \*\* 5 - DECISION; 6 - COUNTER; 7 - MILESTONE;  
 \*\* 8 - PARAMETER CHANGER; 9 - REINITIALIZER;  
 \*\* 15 - THE PSD PHASE TERMINUS  
 \*\*I 1-15: SEPARATE SUBSET ACCUMULATION NUMBER  
 \*\*I 0: NO SEPARATE ACCUMULATION  
 \*\*I 1 - SYSTEM; 3 - CPCL; 4 - CPYC; 5 - MODULE  
 \*\*I THIS BOX PARTICIPATES IN: 1 - DIG REPETITION  
 \*\* 2 - TIC REPETITION; 0 - NEITHER  
 \*\*I 1 - CONTRACTOR; 2 - AIR FORCE; 3 - BOTH

LINE CACI SINSCHRYPT II.5 RELEASE ON 11/09/79 PAGE 2  
 54 FLOW.OVERRIDE(17-10), \*\*I 0 = NORMAL: 1 = STOP  
 55 \*\*I 2 = SKIP THIS BOX  
 56 IN.WT(19-32))  
 57 \*\* INITIAL INTERNAL WAIT TIME BEFORE WORK BEGINS  
 58 \*\* CONSTANT FOR ALL INTEGRATION GROUPS (IGS) AND  
 59 \*\* ALL ITERATIONS  
 60  
 61 \*\*C NUMBER OF ACTIVITY'S PREDECESSORS  
 62 \*\*C POINTER TO 1ST ENTRY IN PREDECESSOR LIST  
 63  
 64 \*\*C NUMBER OF DTS  
 65 \*\*C POINTER TO 1ST ENTRY IN TIMING DATA LIST  
 66  
 67 \*\*C NUMBER OF ACTIVITY'S DIGS OR TIGS, OR 1 IF NONE  
 68 \*\*C POINTER TO 1ST ENTRY IN TIMING & HANPOWER LIST  
 69  
 70 \*\*C NUMBER OF BOX'S "YES" OR SOLE SUCCESSORS  
 71 \*\*C POINTER TO 1ST ENTRY IN SUCCESSOR LIST  
 72  
 73 \*\*C NUMBER OF "NO" OR "LOOP" BRANCH SUCCESSORS  
 74 \*\*C POINTER TO 1ST ENTRY IN "NO" SUCCESSOR LIST  
 75  
 76 \*\*I PROBABILITY OF A "YES" BRANCH, 1ST ITERATION  
 77 \*\*I PROBABILITY OF A "YES" BRANCH, 2ND ITERATION  
 78 \*\*I MILESTONE LABEL CHARACTERS 1-9  
 79  
 80 \*\*I PROBABILITY OF A "YES" BRANCH, 3RD ITERATION  
 81 \*\*I PROBABILITY OF A "YES" BRANCH, 4TH ITERATION  
 82 \*\*I MILESTONE LABEL CHARACTERS 5-9  
 83  
 84 \*\*  
 85 AND ONES  
 86 A PLIST,  
 87 AN OTD,  
 88 AN ATH,  
 89 A YLIST,  
 90 A NLIST  
 91  
 92 \*\*  
 93 DEFINE BOX.ID AS AN ALPHA VARIABLE  
 94 DEFINE LAB1, LAB2 AS ALPHA VARIABLES  
 95  
 96 TEMPORARY ENTITIES  
 97  
 98 EVERY PRED WAS  
 99 A (N.PLIST(1-8),  
 100 S.PLIST(9-32))  
 101 IN WORD 1,  
 102 A (SLA(1-2),  
 103  
 104 PHA(3-3),  
 105  
 106  
 \*\* 1 = ITERATIVE PASS:  
 \*\* 2 = ANY ONE TYPE 3 ENTRY IS NECESSARY TO START  
 \*\* 1 = THIS ENTRY ALONG SUPPLIES TO START  
 \*\* 1 = CURRENT MEMBERSHIP IN THIS SET  
 \*\*C POINTER TO NEXT ENTRY IN THIS SET  
 \*\* 0 = THIS ENTRY ALONG SUPPLIES TO START  
 \*\* 1 = THIS ENTRY IS NECESSARY TO START  
 \*\* 2 = ANY ONE TYPE 3 ENTRY IS NECESSARY TO START  
 \*\* I PROGRESSION MODE ASSIGNED: 0 = FIRST PASS  
 \*\* 1 = ITERATIVE PASS:



```

107 3 = COPY PREDECESSOR'S PROGRESSION MODE
108 0 = NONE; 1 = STEP PREDECESSOR'S DIG DTH.NO;
109 2 = STEP PREDECESSOR'S TIC DTH.NO;
110 3 = COPY DTH.NO FROM PREDECESSOR
111 3 = COPY DTH.NO FROM PREDECESSOR
112 3 = COPY DTH.NO FROM PREDECESSOR
113 3 = COPY DTH.NO FROM PREDECESSOR
114 3 = COPY DTH.NO FROM PREDECESSOR
115 3 = COPY DTH.NO FROM PREDECESSOR
116 3 = COPY DTH.NO FROM PREDECESSOR
117 3 = COPY DTH.NO FROM PREDECESSOR
118 3 = COPY DTH.NO FROM PREDECESSOR
119 3 = COPY DTH.NO FROM PREDECESSOR
120 3 = COPY DTH.NO FROM PREDECESSOR
121 3 = COPY DTH.NO FROM PREDECESSOR
122 3 = COPY DTH.NO FROM PREDECESSOR
123 3 = COPY DTH.NO FROM PREDECESSOR
124 3 = COPY DTH.NO FROM PREDECESSOR
125 3 = COPY DTH.NO FROM PREDECESSOR
126 3 = COPY DTH.NO FROM PREDECESSOR
127 3 = COPY DTH.NO FROM PREDECESSOR
128 3 = COPY DTH.NO FROM PREDECESSOR
129 3 = COPY DTH.NO FROM PREDECESSOR
130 3 = COPY DTH.NO FROM PREDECESSOR
131 3 = COPY DTH.NO FROM PREDECESSOR
132 3 = COPY DTH.NO FROM PREDECESSOR
133 3 = COPY DTH.NO FROM PREDECESSOR
134 3 = COPY DTH.NO FROM PREDECESSOR
135 3 = COPY DTH.NO FROM PREDECESSOR
136 3 = COPY DTH.NO FROM PREDECESSOR
137 3 = COPY DTH.NO FROM PREDECESSOR
138 3 = COPY DTH.NO FROM PREDECESSOR
139 3 = COPY DTH.NO FROM PREDECESSOR
140 3 = COPY DTH.NO FROM PREDECESSOR
141 3 = COPY DTH.NO FROM PREDECESSOR
142 3 = COPY DTH.NO FROM PREDECESSOR
143 3 = COPY DTH.NO FROM PREDECESSOR
144 3 = COPY DTH.NO FROM PREDECESSOR
145 3 = COPY DTH.NO FROM PREDECESSOR
146 3 = COPY DTH.NO FROM PREDECESSOR
147 3 = COPY DTH.NO FROM PREDECESSOR
148 3 = COPY DTH.NO FROM PREDECESSOR
149 3 = COPY DTH.NO FROM PREDECESSOR
150 3 = COPY DTH.NO FROM PREDECESSOR
151 3 = COPY DTH.NO FROM PREDECESSOR
152 3 = COPY DTH.NO FROM PREDECESSOR
153 3 = COPY DTH.NO FROM PREDECESSOR
154 3 = COPY DTH.NO FROM PREDECESSOR
155 3 = COPY DTH.NO FROM PREDECESSOR
156 3 = COPY DTH.NO FROM PREDECESSOR
157 3 = COPY DTH.NO FROM PREDECESSOR
158 3 = COPY DTH.NO FROM PREDECESSOR
159 3 = COPY DTH.NO FROM PREDECESSOR

```

LINE CACI SINCRIPT II.5 RELEASE ON 11/09/79 PAGE 4

```

160 S.ATN(9-32))
161 IN WORD 1,
162 A (BWT1(1/2)).
163 BWT2(2/2)).
164 IN WORD 2,
165 A (ADUB1(1/2)).
166
167
168 ADUB2(2/2))
169 IN WORD 3,
170 A (IDUB1(1/2)).
171 IDUB2(2/2)).
172 IN WORD 4,
173 A (IDUB3(1/2)).
174 IDUB4(2/2)).
175 IN WORD 5,
176
177 A (CHN11(1/4)).
178 CHN12(2/4)).
179 CHN13(3/4)).
180 CHN14(4/4)).
181 IN WORD 6,
182 A (CHN21(1/4)).
183 CHN22(2/4)).
184 CHN23(3/4)).
185 CHN24(4/4)).
186 IN WORD 7,
187 A (CHN31(1/4)).
188 CHN32(2/4)).
189 CHN33(3/4)).
190 CHN34(4/4)).
191 IN WORD 8,
192 A (CHN41(1/4)).
193 CHN42(2/4)).
194 CHN43(3/4)).
195 CHN44(4/4)).
196 IN WORD 9,
197 A (CHN51(1/4)).
198 CHN52(2/4)).
199 CHN53(3/4)).
200 CHN54(4/4)).
201 IN WORD 10,
202 A (CHN11(1/4)).
203 CHN12(2/4)).
204 CHN13(3/4)).
205 CHN14(4/4)).
206 IN WORD 11,
207 A (CHN21(1/4)).
208 CHN22(2/4)).
209 CHN23(3/4)).
210 CHN24(4/4)).
211 IN WORD 12,
212 A (CHN31(1/4)).

```

```

**C POINTER TO NEXT ENTRY IN THIS SET
**S WAIT TO GET ALL NEEDED RESOURCES. ALL ITERATIONS
**S WAIT TO GET ALL NEEDED RESOURCES, 2ND PP
**S SUM OF ACTIVITY DURATIONS, ALL ITERATIONS
** (EXCLUDES PM,PT; 6 EVERY INTER-ITERATION HIATUS,
** IN.WT, 6 WAIT FOR MANPOWER OR OTHER RESOURCES
**S SUM OF ACTIVITY DURATIONS, 2ND PP ITERATIONS
**I ACTIVITY DURATION INPUT
**I ITERATIONS 1 6 2, THIS DTH
**I ACTIVITY DURATION INPUT
**I ITERATIONS 3 6 4PP, THIS DTH
**FOR CHN11 6 CHN11, UNITS ARE 1/16 MEN
**I NUMBER OF PERSONNEL OF TYPE 1 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 1 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 2 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 2 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 3 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 3 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 4 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 4 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 5 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF CONTRACTOR PERSONNEL OF TYPE 5 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF GOVERNMENT PERSONNEL OF TYPE 1 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF GOVERNMENT PERSONNEL OF TYPE 1 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF GOVERNMENT PERSONNEL OF TYPE 2 NEEDED
**I DURING ITERATIONS 1 6 2, THIS DTH
**I NUMBER OF GOVERNMENT PERSONNEL OF TYPE 2 NEEDED
**I DURING ITERATIONS 3 6 4PP, THIS DTH
**I NUMBER OF GOVERNMENT PERSONNEL OF TYPE 3 NEEDED

```

LINE CACT SINCRIPT II.5 RELEASE 8H 11/09/79 PAGE 5  
 213 GRN32(2/4).  
 214 GRN33(3/4).  
 215 GRN34(4/4).  
 216 IN WORD 13.  
 217  
 218 A (GRN11(1/2)).  
 219 GRN12(2/2)).  
 220 IN WORD 14.  
 221 A (GRN21(1/2)).  
 222 GRN22(2/2)).  
 223 IN WORD 15.  
 224 A (GRN31(1/2)).  
 225 GRN32(2/2)).  
 226 IN WORD 16.  
 227 A (GRN41(1/2)).  
 228 GRN42(2/2)).  
 229 IN WORD 17.  
 230 A (GRN51(1/2)).  
 231 GRN52(2/2)).  
 232 IN WORD 18.  
 233 A (GRN61(1/2)).  
 234 GRN62(2/2)).  
 235 IN WORD 19.  
 236 A (GRN71(1/2)).  
 237 GRN72(2/2)).  
 238 IN WORD 20.  
 239 A (GRN81(1/2)).  
 240 GRN82(2/2)).  
 241 IN WORD 21  
 242 AND BELONGS TO AN ATM  
 243 ..  
 244 EVERY YES-SUCC HAS  
 245 A (M-FLIST(1-6)).  
 246 S-FLIST(9-32)).  
 247 IN WORD 1.  
 248 A BOL-POINTER(9-32)  
 249 IN WORD 2  
 250 ADD BELONGS TO A FLIST  
 251 ..  
 252 EVERY NO-SUCC HAS  
 253 A (M-FLIST(1-6)).  
 254 S-FLIST(9-32)).  
 255 IN WORD 1.  
 256 A BOL-POINTER(9-32)  
 257 IN WORD 2.  
 258 A (ALPHA-PAR.  
 259 POINTER-PAR(9-32)).  
 260 REAL-PAR.  
 261 INTEG-PAR(1/2)).  
 262 INTEG-PAR2(2/2)).  
 263 IN WORD 3  
 264 ADD BELONGS TO A FLIST  
 265 ..  
 266  
 267  
 268  
 269  
 270  
 271  
 272  
 273  
 274  
 275  
 276  
 277  
 278  
 279  
 280  
 281  
 282  
 283  
 284  
 285  
 286  
 287  
 288  
 289  
 290  
 291  
 292  
 293  
 294  
 295  
 296  
 297  
 298  
 299  
 300  
 301  
 302  
 303  
 304  
 305  
 306  
 307  
 308  
 309  
 310  
 311  
 312  
 313  
 314  
 315  
 316  
 317  
 318  
 319  
 320  
 321  
 322  
 323  
 324  
 325  
 326  
 327  
 328  
 329  
 330  
 331  
 332  
 333  
 334  
 335  
 336  
 337  
 338  
 339  
 340  
 341  
 342  
 343  
 344  
 345  
 346  
 347  
 348  
 349  
 350  
 351  
 352  
 353  
 354  
 355  
 356  
 357  
 358  
 359  
 360  
 361  
 362  
 363  
 364  
 365  
 366  
 367  
 368  
 369  
 370  
 371  
 372  
 373  
 374  
 375  
 376  
 377  
 378  
 379  
 380  
 381  
 382  
 383  
 384  
 385  
 386  
 387  
 388  
 389  
 390  
 391  
 392  
 393  
 394  
 395  
 396  
 397  
 398  
 399  
 400  
 401  
 402  
 403  
 404  
 405  
 406  
 407  
 408  
 409  
 410  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447  
 448  
 449  
 450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470  
 471  
 472  
 473  
 474  
 475  
 476  
 477  
 478  
 479  
 480  
 481  
 482  
 483  
 484  
 485  
 486  
 487  
 488  
 489  
 490  
 491  
 492  
 493  
 494  
 495  
 496  
 497  
 498  
 499  
 500  
 501  
 502  
 503  
 504  
 505  
 506  
 507  
 508  
 509  
 510  
 511  
 512  
 513  
 514  
 515  
 516  
 517  
 518  
 519  
 520  
 521  
 522  
 523  
 524  
 525  
 526  
 527  
 528  
 529  
 530  
 531  
 532  
 533  
 534  
 535  
 536  
 537  
 538  
 539  
 540  
 541  
 542  
 543  
 544  
 545  
 546  
 547  
 548  
 549  
 550  
 551  
 552  
 553  
 554  
 555  
 556  
 557  
 558  
 559  
 560  
 561  
 562  
 563  
 564  
 565  
 566  
 567  
 568  
 569  
 570  
 571  
 572  
 573  
 574  
 575  
 576  
 577  
 578  
 579  
 580  
 581  
 582  
 583  
 584  
 585  
 586  
 587  
 588  
 589  
 590  
 591  
 592  
 593  
 594  
 595  
 596  
 597  
 598  
 599  
 600  
 601  
 602  
 603  
 604  
 605  
 606  
 607  
 608  
 609  
 610  
 611  
 612  
 613  
 614  
 615  
 616  
 617  
 618  
 619  
 620  
 621  
 622  
 623  
 624  
 625  
 626  
 627  
 628  
 629  
 630  
 631  
 632  
 633  
 634  
 635  
 636  
 637  
 638  
 639  
 640  
 641  
 642  
 643  
 644  
 645  
 646  
 647  
 648  
 649  
 650  
 651  
 652  
 653  
 654  
 655  
 656  
 657  
 658  
 659  
 660  
 661  
 662  
 663  
 664  
 665  
 666  
 667  
 668  
 669  
 670  
 671  
 672  
 673  
 674  
 675  
 676  
 677  
 678  
 679  
 680  
 681  
 682  
 683  
 684  
 685  
 686  
 687  
 688  
 689  
 690  
 691  
 692  
 693  
 694  
 695  
 696  
 697  
 698  
 699  
 700  
 701  
 702  
 703  
 704  
 705  
 706  
 707  
 708  
 709  
 710  
 711  
 712  
 713  
 714  
 715  
 716  
 717  
 718  
 719  
 720  
 721  
 722  
 723  
 724  
 725  
 726  
 727  
 728  
 729  
 730  
 731  
 732  
 733  
 734  
 735  
 736  
 737  
 738  
 739  
 740  
 741  
 742  
 743  
 744  
 745  
 746  
 747  
 748  
 749  
 750  
 751  
 752  
 753  
 754  
 755  
 756  
 757  
 758  
 759  
 760  
 761  
 762  
 763  
 764  
 765  
 766  
 767  
 768  
 769  
 770  
 771  
 772  
 773  
 774  
 775  
 776  
 777  
 778  
 779  
 780  
 781  
 782  
 783  
 784  
 785  
 786  
 787  
 788  
 789  
 790  
 791  
 792  
 793  
 794  
 795  
 796  
 797  
 798  
 799  
 800  
 801  
 802  
 803  
 804  
 805  
 806  
 807  
 808  
 809  
 810  
 811  
 812  
 813  
 814  
 815  
 816  
 817  
 818  
 819  
 820  
 821  
 822  
 823  
 824  
 825  
 826  
 827  
 828  
 829  
 830  
 831  
 832  
 833  
 834  
 835  
 836  
 837  
 838  
 839  
 840  
 841  
 842  
 843  
 844  
 845  
 846  
 847  
 848  
 849  
 850  
 851  
 852  
 853  
 854  
 855  
 856  
 857  
 858  
 859  
 860  
 861  
 862  
 863  
 864  
 865  
 866  
 867  
 868  
 869  
 870  
 871  
 872  
 873  
 874  
 875  
 876  
 877  
 878  
 879  
 880  
 881  
 882  
 883  
 884  
 885  
 886  
 887  
 888  
 889  
 890  
 891  
 892  
 893  
 894  
 895  
 896  
 897  
 898  
 899  
 900  
 901  
 902  
 903  
 904  
 905  
 906  
 907  
 908  
 909  
 910  
 911  
 912  
 913  
 914  
 915  
 916  
 917  
 918  
 919  
 920  
 921  
 922  
 923  
 924  
 925  
 926  
 927  
 928  
 929  
 930  
 931  
 932  
 933  
 934  
 935  
 936  
 937  
 938  
 939  
 940  
 941  
 942  
 943  
 944  
 945  
 946  
 947  
 948  
 949  
 950  
 951  
 952  
 953  
 954  
 955  
 956  
 957  
 958  
 959  
 960  
 961  
 962  
 963  
 964  
 965  
 966  
 967  
 968  
 969  
 970  
 971  
 972  
 973  
 974  
 975  
 976  
 977  
 978  
 979  
 980  
 981  
 982  
 983  
 984  
 985  
 986  
 987  
 988  
 989  
 990  
 991  
 992  
 993  
 994  
 995  
 996  
 997  
 998  
 999  
 1000

```

LINE CACI SIMSCRIPT II.5 RELEASE ON
266 DEFINE ALPHA.PAR AS AN ALPHA VARIABLE
267 DEFINE REAL.PAR AS A REAL VARIABLE
268 **
269 ** GLOBAL VARIABLES
270 ** MONTH OF PROGRAM START (E.G., 12)
271 ** DAY OF PROGRAM START (E.G., 25)
272 ** YEAR OF PROGRAM START (E.G., 1977)
273 ** NUMBER OF MONTHS ACQUISITION PROGRAM IS EXPECTED
274 ** TO LAST (BASIS FOR ARRAY DIMENSIONING)
275 ** NUMBER OF CONTRACTOR MANPOWER TYPES
276 ** NUMBER OF GOVERNMENT MANPOWER TYPES
277 ** NUMBER OF SIMULATION PROGRAM REPEATITIONS
278 ** MAXIMUM NUMBER OF ITERATIONS THROUGH ANY BOX
279 ** NUMBER OF DESIGN GROUPS PER CPCI
280 ** NUMBER OF TEST GROUPS PER CPCI
281 ** NUMBER OF CPACS PER CPCI
282
283 ** AS INTEGER VARIABLES
284 **
285 ** DEFINE DIG.SPREAD AS A 1-DIMENSIONAL ARRAY
286 **
287 ** DEFINE TIG.SPREAD AS A 1-DIMENSIONAL ARRAY
288 **
289 ** DEFINE MANUSE AS A N-DIMENSIONAL INTEGER ARRAY
290 **
291 ** DEFINE MANRATE
292 ** AS A 2-DIMENSIONAL REAL ARRAY
293 **
294 ** DEFINE PLIST,
295 ** OTD,
296 **
297 ** ATR,
298 **
299 ** YLIST,
300 **
301 ** BLIST
302
303 ** AS LIFO SETS
304 ** WITHOUT L, P ATTRIBUTES,
305 ** WITHOUT PL, PB, PA, BL, RS ROUTINES
306 **
307 **
308 ** DEFINE TABLE.A TO BEAN 7
309 ** DEFINE TABLE.B1 TO BEAN 8
310 ** DEFINE TABLE.B2 TO BEAN 9
311 ** DEFINE TABLE.B3 TO BEAN 10
312 ** DEFINE OUTPROPPR TO BEAN 6
313 ** DEFINE USER.DATA TO BEAN 5
314 ** DEFINE INCR TO BEAN ADD 1 TO
315 ** DEFINE DECR TO BEAN SUBTRACT 1 FROM
316 **
317 END

```

11/09/79 PAGE 11

LINE CACI SIMSCRIPT II.5 RELEASE 8H

```

1 ..
2 MAIN
3 ..
4 CALL PROGRAM.FLOW
5
6 PERFORM USER.INPUT
7
8 CALL ORIGIN.R(NORTH.ORIGIN, DAY.ORIGIN, YEAR.ORIGIN)
9
10 PERFORM INPUT.AND.CONVERSION
11
12 END

```

NAME	TYPE	MODE	LINE NUMBERS OF REFERENCES
DAY.ORIGIN	GLOBAL VARIABLE	INTEGER	8
INPUT.AND.CONV	ROUTINE	INTEGER	10
NORTH.ORIGIN	GLOBAL VARIABLE	INTEGER	8
ORIGIN.R	ROUTINE	INTEGER	8
PROGRAM.FLOW	ROUTINE	INTEGER	4
USER.INPUT	ROUTINE	INTEGER	6
YEAR.ORIGIN	GLOBAL VARIABLE	INTEGER	8

11/09/79 PAGE 12

LINE CACI SIMSCRIPT II.5 RELEASE 8H

```

1
2 ROUTINE PROGRAM.FLOW
3 ..
4 .. PREAMBLE
5 .. MAIN
6 .. PROGRAM.FLOW
7 .. USER.INPUT
8 .. ORIGIN.R
9 .. INPUT.AND.CONVERSION
10 .. BOX.COUNT
11 .. INIT.TABLE.A
12 .. GET.BOX.INFO
13 .. ATN.CREATION
14 .. OTD.CREATION
15 .. INIT.TABLE.A
16 .. READ.PREDS.AND.SUCS
17 .. PRED.CREATION
18 .. YES.SUC.CREATION
19 .. NO.SUC.CREATION
20 .. SIMULATION
21 .. ANALYSIS
22 ..
23 RETURN
24 END

```

.. OF ROUTINE PROGRAM.FLOW

```

1 ROUTINE FOR USER.INPUT
2 **
3 ** USER.INPUT READS IN GLOBAL VARIABLES THAT WILL BE CHANGED
4 ** OFTEN BETWEEN EXECUTIONS.
5 **
6 ** ROUTINE LAST MODIFIED: NOVEMBER 9, 1979
7 **
8
9 DEFINE INDEX
10 AS AN INTEGER VARIABLE
11
12 USE USER.DATA FOR INPUT
13 USE OUTBUFFER FOR OUTPUT
14 LET EOP.V = 1
15
16 READ MONTH.ORIGIN, DAY.ORIGIN, YEAR.ORIGIN
17 WRITE MONTH.ORIGIN, DAY.ORIGIN, YEAR.ORIGIN AS
18 "DATE OF PROGRAM START: ",I 2,"/",I 2,"/",I 4, /
19
20 READ MONTHS, REPERITIONS, IT.LIMIT
21 WRITE MONTHS, REPERITIONS, IT.LIMIT AS
22 "THE ACQUISITION PROGRAM IS EXPECTED TO LAST ",I 4," MONTHS. ",
23 "/. "THE SIMULATION PROGRAM WILL HAVE ",I 4," REPERITIONS."/.,
24 "THERE WILL BE NO MORE THAN ",I 4," ITERATIONS THROUGH",
25 " ANY BOX."/
26
27 READ C.NANTYPES, G.NANTYPES
28 WRITE C.NANTYPES, G.NANTYPES AS "THE NUMBER OF CONTRACTOR ",
29 "HARPONER TYPES IS: ",I 4,"/,"THE NUMBER OF GOVERNMENT ",
30 "HARPONER TYPES IS: ",I 4, /
31
32 READ MDIG
33 RESERVE DIG.SPREAD AS MDIG
34 READ DIG.SPREAD
35
36 READ MTIG
37 RESERVE TIC.SPREAD AS MTIG
38 READ TIC.SPREAD
39
40 READ MCPRC
41
42 WRITE MDIG AS "FOR ",I 2," DIGS, THE BREAKDOWNS ARE: "
43 FOR INDEX = 1 TO MDIG, DO
44 WRITE DIG.SPREAD(INDEX) AS I 3, "%", S 8
45 LOOP
46
47 WRITE MTIG AS "/. "FOR ",I 2," TIGS, THE BREAKDOWNS ARE: "
48 FOR INDEX = 1 TO MTIG, DO
49 WRITE TIC.SPREAD(INDEX) AS I 3, "%", S 8
50 LOOP
51
52 WRITE MCPRC AS "/. "THERE ARE ", I 3, " CPARC'S", /
53
54
55 RETURN
56 END

```

```

LINE CACI SIMSCRIPT II.5 RELEASE 8H
1 ROUTINE INPUT.AND.CONVERSION
2 **
3 ** INPUT.AND.CONVERSION IS THE CONTROL ROUTINE FOR READING IN ALL
4 ** INPUT DATA. CONVERSIONS TO THE INPUT DATA ARE PERFORMED IN OTHER
5 ** ROUTINES BEFORE IT IS PUT INTO THE DATA BASE.
6 **
7 ** ROUTINE LAST MODIFIED: 9-24-79
8
9 DEFINE HEADING TO MEAN 7 ** NUMBER OF LINES IN TABLE A HEADING
10 DEFINE I AS AN INTEGER VARIABLE ** LOOP SUBSCRIPT
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2
```

```

1
2
3
4 ROUTINE TO INIT.TABLE.A
5
6 ** INIT.TABLE.A IS USED TO RE-INITIALIZE INPUT TABLE A TO BEGIN
7 ** READING AT THE TOP OF THE FILE. THE LINES THAT MAKE UP THE
8 ** HEADING FOR THE TABLE ARE AUTOMATICALLY SHIPPED AND THE END OF FILE
9 ** STATUS INDICATOR IS RESET.
10
11 ** ROUTINE LAST MODIFIED: 9-24-79
12
13
14
15
16
17
18
19
20
21
22
23
24

```

```

24 END ** ROUTINE INIT.TABLE.A

```

```

1 ROUTINE FOR BOX.COUNT
2
3 **
4 ** BOX.COUNT READS THROUGH TABLE A TO OBTAIN A TOTAL OF BOXES TO
5 ** USE IN THE CREATE STATEMENT FOR THE PERMANENT ENTITY 'BOX'.
6
7 ** NOTE: AT THIS TIME A MAXIMUM OF 25 BOXES ARE USED
8
9 ** ROUTINE LAST MODIFIED: 9-24-79
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

```



```

1 ROUTINE TO GET BOX INFO
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

THIS ROUTINE READS THE DATA COMMON TO ALL TYPES OF BOXES AND  
 CONVERTS IT TO INTEGER FORMAT BEFORE PUTTING IT INTO THE DATA BASE.  
 NOTE: SUBNETWORK DATA HAS NOT YET BEEN INCLUDED IN TABLE A  
 (9-24-79). ALL REFERENCES TO IT IN THIS ROUTINE HAVE  
 BEEN COMMENTED OUT.  
 ROUTINE LAST MODIFIED: 11-5-79

```

    DEFINE ID,
    TYPE,
    GROUP,
    AND WHO
    AS ALPHA VARIABLES

    DEFINE SUBNET
    AND I
    AS INTEGER VARIABLES

    FOR I = 1 TO N BOX, DO
      READ ID, TYPE, NON-LINES, WHO, LEVEL, GROUP, SUBNET
      WRITE ID, TYPE, NON-LINES, WHO, LEVEL, AND GROUP AS A 4, S 3,
        I 1, I 5, S 4, A 1, I 6, S 4, A 1, S 10,
        "IS CONVERTED TO", /

      LET BOX.ID(I) = ID

    ** CONVERT BOX TYPE FROM ALPHA (VARIABLE NAME ID) TO
    ** INTEGER FORMAT

    IF TYPE = "A"
      LET BOX.TYPE(I) = 2
    ELSE IF TYPE = "B"
      LET BOX.TYPE(I) = 5
    ELSE IF TYPE = "C"
      LET BOX.TYPE(I) = 6
    ELSE IF TYPE = "H"
      LET BOX.TYPE(I) = 3
    ELSE IF TYPE = "P"
      LET BOX.TYPE(I) = 7
    ELSE IF TYPE = "S"
      LET BOX.TYPE(I) = 4
    ELSE WRITE TYPE AS
      "*****ERROR -> INVALID BOX TYPE: ",
      A 3, /

    ALWAYS
    ALWAYS
  
```

```

LINE CACI SIMSCRIPT II.5 RELEASE 84
54         ALWAYS
55         ALWAYS
56         ALWAYS
57
58         LET SUBNETWORK(I) = SUBNET
59
60         LET LEVEL(I) = LEVEL
61
62         **CONVERT DTM FROM ALPHA (VARIABLE NAME GROUP) TO INTEGER FORMAT
63
64         IF GROUP = "D"
65             LET DTM(I) = 1
66         ELSE IF GROUP = "T"
67             LET DTM(I) = 2
68         ELSE IF GROUP = "B"
69             LET DTM(I) = 3
70         ELSE WRITE GROUP AS "*****ERROR --> INVALID DTM: ",
71             A 3, /, /
72         ALWAYS
73         ALWAYS
74
75         ** CONVERT DOER FROM ALPHA (VARIABLE NAME WHO) TO INTEGER
76
77         IF WHO = "A"
78             LET DOER(I) = 2
79         ELSE IF WHO = "C"
80             LET DOER(I) = 1
81         ELSE IF WHO = "B"
82             LET DOER(I) = 3
83         ELSE IF WHO = "H"
84             LET DOER(I) = 0
85         ELSE WRITE WHO AS "*****ERROR --> INVALID DOER: ",
86             A 3, /, /
87         ALWAYS
88         ALWAYS
89         ALWAYS
90
91         ALWAYS
92
93         WRITE BOX.ID(I), BOX.TYPE(I), DOER(I),
94             LEVEL(I), AND DTM(I) AS A 4, I 4, I 10, I 6,
95             I 5, /, /
96
97         START NEW CARD
98         SKIP (SUB.LINES - 1) CARDS
99
100         IF BOX.TYPE(I) = 2
101             PERFORM ATH.CREATION GIVEN I
102         ALWAYS
103
104         PERFORM OTD.CREATION
105
106     LOOP

```

```

1 ROUTINE TO READ.PREDS.AND.SUCS
2 **
3 ** THIS ROUTINE CONTROLS THE PREDECESSOR AND SUCCESSOR CREATION
4 ** ROUTINES USING THE QUANTITY COUNTS FOR EACH TYPE OF ENTITY.
5 **
6 ** ROUTINE LAST MODIFIED: 9-24-79
7 **
8
9 DEFINE INDEX, ** ARRAY INDEX FOR BOX INFORMATION
10 QNTS, ** NUMBER OF SUCCESSORS FOR 'NO' DECISION
11 QNTS, ** NUMBER OF SUCCESSORS FOR 'YES' DECISION
12 QNTS, ** NUMBER OF PREDECESSORS
13 QNTS, ** NUMBER OF PREDECESSORS
14 AND NUM.LINES ** NUMBER OF LINES OF INFORMATION FOR THIS
15 AS INTEGER VARIABLES ** BOX IN TABLE A.
16
17 WRITE AS %, "READING IN PREDECESSORS BEGINS HERE", //
18 FOR INDEX = 1 TO W.BOX, DO
19
20     SKIP 2 FIELDS
21     READ NUM.LINES
22     SKIP 3 FIELDS
23     READ QNTS.PRED
24     WRITE BOX.ID(INDEX) AND QNTS.PRED AS /, S 90, A 4, " WAS ",
25         I 4, " PREDECESSOR(S) ", //
26
27     IF QNTS.PRED > 0
28         PERFORM PRED.CREATION GIVEN INDEX
29         DECR QNTS.PRED
30         ALWAYS
31
32     READ QNTS, QNTS
33     WRITE QNTS AND QNTS AS S 100, I 3, " NO SUCCESSOR(S) ", //,
34         S 100, I 3, " YES SUCCESSOR(S) ", /
35
36     IF QNTS > 0
37         PERFORM YES.SUC.CREATION GIVEN INDEX
38         DECR QNTS
39         ALWAYS
40
41     IF QNTS > 0
42         PERFORM NO.SUC.CREATION GIVEN INDEX
43         DECR QNTS
44         ALWAYS
45
46     DECR NUM.LINES
47     START NEW CARD
48
49     UNTIL NUM.LINES = 0, DO
50
51         IF QNTS.PRED > 0
52             PERFORM PRED.CREATION GIVEN INDEX
53             DECR QNTS.PRED

```

```

LINE CACI SINSCHRIPT II.5 RELEASE 8H
54      ALWAYS
55
56      IF QNYS > 0
57          PERFORM YES.SUC.CREATION GIVEN INDEX
58          DECR QNYS
59      ALWAYS
60
61      IF QNN > 0
62          PERFORM NO.SUC.CREATION GIVEN INDEX
63          DECR QNN
64      ALWAYS
65
66      DECR NON.LINES
67      START NEW CARD
68      SKIP 1 LINE
69
70      LOOP
71
72      RETURN
73
74  END **OF ROUTINE TO READ.PREDS.AND.SUCS

```

```

1
2
3 ROUTINE FOR PRED.CREATION GIVEN BOX.INDEX
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

THIS ROUTINE CREATES A PREDECESSOR ENTITY, READS IN AND CONVERTS  
 THE PREDECESSOR DATA, AND FILES THE PREDECESSOR ON THE APPROPRIATE  
 PREDECESSOR LIST.

ROUTINE LAST MODIFIED: 9-24-79

DEFINE ID.                    .. PREDECESSOR NAME  
 EXIT,                        .. TYPE OF EXIT LEADING INTO PREDECESSOR  
 GROUP,                       .. DTM IN ALPHA  
 PROG.MODE,                  .. PROGRESSION MODE IN ALPHA  
 AND START                    .. START LOGIC - SLA IN ALPHA  
 AS ALPHA VARIABLES

DEFINE BOX.INDEX            .. INDEX TO WHICH THIS PREDECESSOR BELONGS  
 AND K                        .. LOOP INDEX  
 AS INTEGER VARIABLES

CREATE A PRED

READ ID, EXIT, GROUP, PROG.MODE, START

\*\* CONVERT ALPHA 'GROUP' TO INTEGER 'DTM'

IF GROUP = "M"  
 LET GP.CTR1(PRED) = 0  
 ELSE IF GROUP = "C"  
 LET GP.CTR1(PRED) = 3  
 ELSE IF GROUP = "D"  
 LET GP.CTR1(PRED) = 1  
 ELSE IF GROUP = "I"  
 LET GP.CTR1(PRED) = 2  
 ELSE WRITE GROUP AS "\*\*\*\*\*ERROR -> INVALID",  
 " GROUP: ", A 3, //

ALWAYS  
 ALWAYS  
 ALWAYS

\*\* CONVERT ALPHA 'PROG.MODE' TO INTEGER PHA

IF PROG.MODE = "T"  
 LET PHA(PRED) = 0  
 ELSE IF PROG.MODE = "I"  
 LET PHA(PRED) = 1  
 ELSE IF PROG.MODE = "C"  
 LET PHA(PRED) = 3  
 ELSE WRITE PROG.MODE AS "\*\*\*\*\*ERROR -> INVALID",  
 " PROGRESSION MODE: ", A 4, //

ALWAYS

```

LINE CACT SIMSCRIPT II.5 RELEASE ON
54     ALWAYS
55     ALWAYS
56
57     ** CONVERT ALPHA 'START' TO INTEGER SLA
58
59     IF START = "M"
60     LET SLA(PRED) = 0
61     ELSE IF START = "A"
62     LET SLA(PRED) = 1
63     ELSE IF START = "R"
64     LET SLA(PRED) = 2
65     ELSE WRITE START AS "*****ERROR -> INVALID START",
66         " LOGIC: ", A 3, //
67     ALWAYS
68     ALWAYS
69
70     ** GET PREDECESSOR'S ARRAY INDEX
71
72     FOR K = 1 TO N.BOX, WITH BOX.ID(N) = ID,
73     FIND BOX.POINTER(PRED) = K, IF NONE WRITE AS
74     "----WARNING -> PREDECESSOR NOT INCLUDED ON ",
75     "BOX.ID LIST "
76     ALWAYS
77
78     FILE PRED FIRST IN PLIST(BOX.INDEX)
79
80     WRITE BOX.ID(BOX.INDEX) AND ID AS A N,
81     " HAS PREDECESSOR ", A N, //
82     RETURN
83
84     END
85     **ROUTINE PRED.CREATION

```

```

1 ROUTINE FOR YES.SUC.CREATION GIVEN BOX.INDEX
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

```

\*\* THIS ROUTINE CREATES A SUCCESSOR THAT FOLLOWS FROM EITHER A  
 YES DECISION ON A BRANCH BOX OR A STRAIGHTFORWARD SUCCESSION. THE  
 SUCCESSOR DATA IS READ IN, CONVERTED TO INTEGER FORMAT, AND FILED  
 ON THE APPROPRIATE SUCCESSOR LIST.  
 \*\* ROUTINE LAST MODIFIED: 9-24-79  
 \*\* BOX NAME  
 AS AN ALPHA VARIABLE  
 DEFINE BOX.INDEX \*\* BOX TO WHICH THIS SUCCESSOR BELONGS  
 AND I \*\* LOOP INDEX  
 AS INTEGER VARIABLES  
 CREATE A YES.SUCC  
 READ ID  
 \*\* GET SUCCESSOR'S ARRAY INDEX  
 FOR I = 1 TO N.BOX, WITH BOX.ID(I) = ID,  
 FIND BOX.POINTER(YES.SUCC) = 1,  
 IF MORE WRITE AS  
 "----WARNING --> YES SUCCESSOR NOT INCLUDED ON BOX.ID LIST "  
 ALWAYS  
 FILE YES.SUCC FIRST IN YLIST(BOX.INDEX)  
 WRITE BOX.ID(BOX.INDEX) AND ID AS A 4, " HAS YES SUCCESSOR ",  
 A 4, /  
 RETURN  
 END \*\*\*OF ROUTINE FOR YES.SUC.CREATION

```

1
2 ROUTINE FOR NO.SUC.CREATION GIVEN BOX.INDEX
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

THIS ROUTINE CREATES THE SUCCESSORS THAT RESULT FROM A NO  
 DECISION ON A BRANCH BOX. THE SUCCESSOR DATA IS READ IN, CONVERTED  
 TO INTEGER FORMAT, AND FILED ONTO THE APPROPRIATE SUCCESSOR LIST.  
 ROUTINE LAST MODIFIED: 9-24-79

```

    DEFINE ID
    AS AN ALPHA VARIABLE
    DEFINE BOX.INDEX
    AND I
    AS INTEGER VARIABLES
    CREATE A NO.SUCC
    READ ID
    FIND SUCCESSOR'S ARRAY INDEX
    FOR I = 1 TO N BOX, WITH BOX.ID(I) = ID,
    FIND BOX.POINTER(NO.SUCC) = I,
    IF MORE WRITE AS "----WARNING --> NO SUCCESSOR NOT INCLUDED",
    " ON BOX.ID LIST "
    ALWAYS
    FILE NO.SUCC FIRST IN MLIST(BOX.INDEX)
    WRITE BOX.ID(BOX.INDEX) AND ID AS A 4, " HAS NO SUCCESSOR ",
    A 4, /
    RETURN
  
```

END \*\*OF ROUTINE FOR NO.SUC.CREATION



DATE OF PROGRAM START: 11/ 1/1979  
 THE ACQUISITION PROGRAM IS EXPECTED TO LAST 24 MONTHS.  
 THE SIMULATION PROGRAM WILL HAVE 15 REPETITIONS.  
 THERE WILL BE NO MORE THAN 4 ITERATIONS THROUGH ANY BOX.  
 THE NUMBER OF CONTRACTOR HANDPOWER TYPES IS: 6  
 THE NUMBER OF GOVERNMENT HANDPOWER TYPES IS: 2  
 FOR 2 DIGS, THE BREAKDOWNS ARE: 45% 30%  
 FOR 3 DIGS, THE BREAKDOWNS ARE: 65% 30%  
 THERE ARE 6 CPNC'S 5%

NO.	TYPE	LMS	DOEN	LVL	GRP	SUBNET
2A	A	2	C	1	M	IS CONVERTED TO
2A	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
3A	A	4	C	1	M	IS CONVERTED TO
3A	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
4C	A	2	B	1	M	IS CONVERTED TO
4C	2			3	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
5E	A	1	A	1	M	IS CONVERTED TO
5E	2			2	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
6S	B	1	A	1	M	IS CONVERTED TO
6S	5			2	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
7J	A	2	B	1	M	IS CONVERTED TO
7J	2			3	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
8L	B	1	A	1	M	IS CONVERTED TO
8L	5			2	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
9H	A	1	C	1	M	IS CONVERTED TO
9H	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
9S	A	3	C	1	M	IS CONVERTED TO
9S	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
6A	A	3	C	3	M	IS CONVERTED TO
6A	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----
6D	A	1	C	3	M	IS CONVERTED TO
6D	2			1	0	
						----- IN ROUTINE FOR ATH.CREATION -----
						----- IN ROUTINE FOR OTD.CREATION -----

6E	B	1	C	3	M		IS CONVERTED TO	
6E	5		1	3	0			
6F	A	2	C	3	M		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
6F	2		1	3	0			
6G	A	5	C	3	D		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
6G	2		1	3	1			----- IN ROUTINE FOR OTD.CREATION -----
6H	A	1	C	3	D		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
6H	2		1	3	1			----- IN ROUTINE FOR OTD.CREATION -----
6I	B	2	C	3	D		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
6I	5		1	3	1			----- IN ROUTINE FOR OTD.CREATION -----
6L	E	1	M	3	M		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
6L	7		0	3	0			
6M	A	2	C	3	D		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
6M	2		1	3	1			
6P	B	1	B	3	B		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
6P	5		3	3	0			----- IN ROUTINE FOR OTD.CREATION -----
9A	A	2	B	3	D		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
9A	2		3	3	1			
9C	B	2	A	3	D		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
9C	5		2	3	1			----- IN ROUTINE FOR OTD.CREATION -----
9E	C	1	B	3	D		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
9E	6		3	3	1			
10A	A	4	C	4	D		IS CONVERTED TO	----- IN ROUTINE FOR OTD.CREATION -----
10A	2		1	4	1			
10C	A	1	C	3	D		IS CONVERTED TO	----- IN ROUTINE FOR ATH.CREATION -----
								----- IN ROUTINE FOR OTD.CREATION -----



READING IN PREDECESSORS BEGINS HERE

2A	HAS	0 PREDECESSOR(S) 0 NO SUCCESSOR(S) 0 YES SUCCESSOR(S)
4A	HAS	2 PREDECESSOR(S) 0 NO SUCCESSOR(S) 4 YES SUCCESSOR(S)
4C	HAS	2 PREDECESSOR(S) 0 NO SUCCESSOR(S) 1 YES SUCCESSOR(S)
4E	HAS	1 PREDECESSOR(S) 0 NO SUCCESSOR(S) 1 YES SUCCESSOR(S)
4G	HAS	1 PREDECESSOR(S) 1 NO SUCCESSOR(S) 1 YES SUCCESSOR(S)
4J	HAS	2 PREDECESSOR(S) 0 NO SUCCESSOR(S) 1 YES SUCCESSOR(S)
4L	HAS	1 PREDECESSOR(S) 1 NO SUCCESSOR(S) 1 YES SUCCESSOR(S)

2A HAS PREDECESSOR 2A

4A HAS YES SUCCESSOR 4C  
4A HAS PREDECESSOR 4E  
4A HAS YES SUCCESSOR 4A

---WARNING --> YES SUCCESSOR NOT INCLUDED ON BOX.ID LIST 4A HAS YES SUCCESSOR 53A  
---WARNING --> YES SUCCESSOR NOT INCLUDED ON BOX.ID LIST 4A HAS YES SUCCESSOR 53C

4C HAS PREDECESSOR 4A

4E HAS YES SUCCESSOR 4E  
4C HAS PREDECESSOR 4G

4E HAS PREDECESSOR 4C

4E HAS YES SUCCESSOR 4G

4G HAS PREDECESSOR 4E

4G HAS YES SUCCESSOR 4J  
4G HAS NO SUCCESSOR 4C

4J HAS PREDECESSOR 4G

4J HAS YES SUCCESSOR 4L  
4J HAS PREDECESSOR 4A

4L HAS PREDECESSOR 4J

---WARNING --> YES SUCCESSOR NOT INCLUDED ON BOX.ID LIST 4L HAS YES SUCCESSOR 66B



52	HAS PREDECESSOR 62				
53	HAS PREDECESSOR 61				
54	HAS PREDECESSOR				
55	HAS PREDECESSOR 02				
6H	HAS	1 PREDECESSOR(S)			
		0 NO SUCCESSOR(S)			
		1 YES SUCCESSOR(S)			
6I	HAS	1 PREDECESSOR(S)			
		1 NO SUCCESSOR(S)			
		2 YES SUCCESSOR(S)			
6L	HAS	1 PREDECESSOR(S)			
		0 NO SUCCESSOR(S)			
		0 YES SUCCESSOR(S)			
6M	HAS	1 PREDECESSOR(S)			
		0 NO SUCCESSOR(S)			
		2 YES SUCCESSOR(S)			
6P	HAS	1 PREDECESSOR(S)			
		1 NO SUCCESSOR(S)			
		1 YES SUCCESSOR(S)			
8A	HAS	2 PREDECESSOR(S)			
		0 NO SUCCESSOR(S)			
		1 YES SUCCESSOR(S)			
8C	HAS	1 PREDECESSOR(S)			
		1 NO SUCCESSOR(S)			

56 HAS YES SUCCESSOR 8A  
 61 HAS NO SUCCESSOR 6G  
 ---JABBING --> YES SUCCESSOR NOT INCLUDED ON BOX.ID LIST 6I HAS YES SUCCESSOR 20A

6L HAS PREDECESSOR 8C  
 6M HAS PREDECESSOR 8C  
 5M HAS YES SUCCESSOR 02  
 6M HAS YES SUCCESSOR 10A

6P HAS PREDECESSOR 8C  
 6P HAS YES SUCCESSOR 6A  
 6P HAS NO SUCCESSOR 6C

8A HAS PREDECESSOR 6I  
 8A HAS YES SUCCESSOR 8C  
 ---JABBING --> PREDECESSOR NOT INCLUDED ON BOX.ID LIST 8A HAS PREDECESSOR 20E

8C HAS PREDECESSOR 8A





PREDECESSOR SUMMARY		# PREDs
BOX		
2A		0
4A		2
4C		2
4E		1
4G		1
4J		2
4L		1
4H		1
4S		1
6A		3
6D		1
6E		1
6F		1
6G		5
6H		1
6I		1
6L		1
6H		1
6P		1
8A		2
8C		1
8E		1
10A		4
10C		1
10E		1

APPENDIX F  
EVENT NOTICE SPECIFICATION

1. INTRODUCTION

In the Simulation Conduct Processor the overall control of flow through the network will be accomplished by the use of SIMSCRIPT II.5 Event Notices to control the box-to-box succession and to pass parameters. Two principal types of Event Notice are required; each has its own parameter list and an associated Event Routine to accomplish its tasks. These Event Types, listed below, are fully specified subsequently. Other (and simpler) Event Types (e.g., to end each simulation pass, to end simulation) will be specified during FY 80.

<u>Para</u>	<u>Name</u>	<u>Function</u>
3.1	Flow	<p>The Flow Event Type provides a standard mechanism for transitioning from a terminating box to its successor boxes. A Flow Event is scheduled whenever a Box.Proc Event Routine has completed its processing on the designated box. When invoked, the Flow Event Routine performs termination processing on the completed box and then performs the following actions (as applicable) for each successor box:</p> <ul style="list-style-type: none"><li>- Determination of Group Number and Progression Mode;</li><li>- Predecessor Wait and Internal Wait processing;</li><li>- Predecessor Input Status updating; and</li><li>- Box.Proc Event Notice scheduling.</li></ul>
3.2	Box.Proc	<p>The Box.Proc Event Type initiates a common process for simulating appropriate actions whenever a process function box is ready to begin. The associated Event Routine observes special override instructions, if any, then determines the box type, and calls in the corresponding box processor routine. When the box type specific actions have been completed, a Flow Event is scheduled for the box.</p>

2. DATA NOTATION

Each of the following Event Routine descriptions makes extensive reference to the Global Data Base; this is defined (as yet incompletely) in the coded program's Preamble, as listed in Appendix E. While the Event Routines use the listing data names, additional notation is provided below to convey the dimensionality of much of the data. This notation, which is described below, is intended as an aid to the reader.

- a. Each data name is shown within parenthesis.
- b. All underlined data items are variables; all other items remain constant during the simulation run.
- c. Dimensional factors are indicated by small letters which immediately follow the enclosed name, with commas being used as separators. The following factors are used:

g = Integration Group number (DTN.No) dependency

i = iteration number (It.Ct) dependency

a = assigned manpower type; i.e., one of a vector of manpower quantities needed for each task.

m = monthly accumulation of data

p = predecessor list membership

s = successor list membership.

- d. Several examples illustrate this usage.

(CMN)a,g,i

"CMN" contains the contractor's manpower assignment.

"a" indicates that each value belongs to a vector of manning levels for the different assigned types of personnel.

"g" indicates that the duration may be different for each Integration Group (DTN.No).

"i" indicates that the manpower level can change per iteration.

(IDUR)g,i

"IDUR" contains the duration assigned to an activity.

"g" indicates that the duration may be different for each Integration Group (DTN.No).

"i" indicates that the duration is also dependent on the iteration count (It.Ct) for the Group.

(PMN)p,g

"PMN" indicates the Predecessor Input Status for a box.

"p" indicates that each PMN is part of a vector which is ordered on predecessor list position.

"g" indicates that Predecessor Input Status is maintained for each Integration Group.

(FNETCM)a,m

"FNETCM" accumulates the contractor manpower expended on the Full network.

"a" indicates that each value belongs to a vector per assigned personnel type.

"m" indicates that the data are aggregated into monthly packets.

### 3. EVENT NOTICE & COMPUTER ROUTINE SPECIFICATION

#### 3.1 Flow: Network Flow Effector

##### 3.1.1 Purpose

The Flow Event Program performs all necessary box termination activities for the designated box and then updates the Predecessor Input Status for each box listed in the designated set of successors. It also performs Internal Wait processing and Event Notice scheduling for each successor (if any) whose predecessor input conditions have been met.

##### 3.1.2 Event Scheduling

A Flow Event Notice is scheduled for a designated box whenever the Box.Proc Event Routine completes its processing of that box.

##### 3.1.3 Actions

###### a. Box Termination Processing

(1) The following incoming parameters are saved as local variables:

- (a) Box.Point becomes Own.Box: the pointer to the box to be terminated.
- (b) PMN becomes PMP: the Progression Mode of the predecessor box.
- (c) DTNN becomes PDTN: The Integration Group (i.e., DTN.No) of the predecessor.

- (d) XIT is used to select the designated list of successor boxes (YList or NList).
- (2) Own.Box Status (Box.Stat)g is set to zero (OFF).
- (3) For Activity Boxes only, all previously allocated resources are released. If an override was in effect for the box (Flow.Override NEq 0), no resources were used; therefore no resources need to be released. Otherwise, the following actions are taken:
  - (a) Each element of the contractor manpower use vector ((CMN)a,i) is subtracted from the corresponding network total ((CMNA)a).
  - (b) Each element of the government manpower use vector ((GMN)a,i) is subtracted from the corresponding network total ((GMNA)a).
  - (c) Both of the above manpower use network total vectors are saved in the sequential activity history file (Rsr.History). This file, which will contain a timed accounting of all resource transactions, will support the creation of a resource utilization profile. In later versions the impacts resulting from resource limitations will be simulated.

b. Successor Box (S-Box) Processing. For each successor box (S-Box) in the Own.Box successor list designated by input Event Notice parameter XIT, the following actions are effected; if the successor list is empty, Flow returns with no further action.

- (1) The successor list Box.Pointer is used to gain access to the following S-Box data: the Box.Type; its Group dependency (DTN); and its list of predecessors (PList).
- (2) The S-Box predecessor list (PList) is searched to find the Own.Box entry (Plist Box.Pointer = Own.Box). The following data are extracted:
  - (a) "p" - Own.Box position on the PList
  - (b) (SLA)p - Start Logic Assignment (see Appendix A-1)
  - (c) (Gp.Ctrl)p - Group Number (DTN.No) Controller (see Appendix A-1)
  - (d) (PMA)p - Progression Mode Assignment (see Appendix A-1)
  - (e) "n" - Own.Box location on the Predecessor Input Status Indicator vector.

(3) The Integration Group number (DTN.No) for this Predecessor Input update is then obtained as follows:

- (a) If Own.Box Group Dependency (DTN) is "none"(0), DTN.No is set to 1; otherwise:
- (b) The value of (Gp.Ctrl)<sub>p</sub> determines the Integration Group number (DTN.No) as follows:

<u>Gp.Ctrl</u>	<u>DTN.No =</u>	<u>Comment</u>
0	1	(S-Box not group dependent)
1	PDTN+1	(Increment DIG Count)
2	PDTN+1	(Increment TIG Count)
3	PDTN	(Continue Pred DTN.No)

(4) The Progression Mode (PMN) is then determined using the S-Box (PMA)<sub>p</sub> and the local (PMP); as follows:

<u>PMA</u>	<u>PMN =</u>	<u>Comment</u>
1	1	Forward Progression mandated
2	2	Iterative Progression mandated
3	PMP	Continue Pred Progression

(5) If PMN = 2 (Iterative Progression) the action continues at paragraph 3.1.3b(11), Iteration Count Actions.

(6) If PMN = 1 (Forward Progression) and SLA = 0 (Single Input Start), the Predecessor Wait (Pr.Wt)<sub>g</sub> is set to zero, and the action continues at paragraph 3.1.3b(11), Iteration Count Actions.

(7) For all other Forward Progression cases, the Pred input status is updated per paragraph 3.1.3b(8); the actions beyond this point depend on the results of the Pred Input Status Check in paragraph 3.1.3b(9).

(8) Pred Input Status Update

- (a) If (PDNF)<sub>g</sub> = 0 (No prior Pred inputs yet received), set (PDNF)<sub>g</sub> = 1 and set (Pr.Wt)<sub>g</sub> = TIME.V. (TIME.V is the current Simulation Time.) The latter action saves the starting time for the Predecessor Wait (Pr.Wt)<sub>g</sub> calculation per paragraph 3.1.3b(10). The box is then placed in "Pred Wait" status (Box.Stat = 1).
- (b) If (SLA)<sub>p</sub> = 2 (Or-type input), set (PDNR)<sub>g</sub> = 1 (logical OR input is satisfied). (Note: (PDNR)<sub>g</sub> may already = 1, in which case it will remain 1).
- (c) If (SLA)<sub>p</sub> = 1 (And-type input), Set (PDN)<sub>n</sub> = 1.

(9) Pred Input Status Dependent Actions

If all Pred Input Status conditions are found to be satisfied (i.e.,  $(PDNR)g = 1$  and  $(PDN)n,g = 1$  for all values of "n") for an S-Box, the box is set up for activation per paragraphs 3.1.3b(10) - (13); otherwise the program returns.

(10) Set Pred Wait (Pr.Wt) as follows:

$(Pr.Wt)g = TIME.V - (Pr.Wt)g$ . (See paragraph 3.1.3b(8)(a)).

(11) Iteration Count (It.Ct)g Actions

(a) Step (It.Ct)g

- (b) If (It.Ct)g remains within limits (i.e., is less than It.Limit), continue with paragraph 3.1.3b(12); otherwise the following error message is output, and exit occurs.

IT.CT HIGH, (Box.ID), (DTN.No), (TIME.V)

(12) Internal Wait Processing

Internal Waits reflect inherent delays in starting a function (e.g., the time between when a document is ready for submittal until it is in the hands of the reviewers). It therefore applies each time the box is entered. If no Internal Wait is required ( $In.Wt = 0$ ) the Event Notice time (TIME.A) is set to "now" (TIME.V) and the action skips to paragraph 3.1.3b(13). Otherwise:

- (a) The Wait Duration (I.Wait) is set equal to the box parameter In.Wt. In future Simulator versions, In.Wt will be subjected to random perturbation before each use.
- (b) The Wait is then added to the Cumulative Internal Wait;  $(CIn.Wt)g = (CIn.Wt)g + (I.Wait)$ .
- (c) The box is put into Internal Wait status;  $(Box.Stat)g = 2$ .
- (d) If the Progression Mode is Forward ( $PMN = 0$ ) the Earliest Start Time is set to now ( $(EST)g = TIME.V$ ).
- (e) The Event Notice time (TIME.A) is set to its value for the end of the Wait ( $TIME.V + I.Wait$ ).

(13) Event Notice Generation

A Box.Proc Event Notice is created and scheduled as follows:

- (a) The Event Time (TIME.A) contains the value established in paragraph 3.1.3b(12).

- (b) The Box.Point designates the S-Box being processed (i.e., (Box.Pointer)s).
- (c) The Integration Group number (DTNN) is set to the value of DTN.No as obtained in paragraph 3.1.3b(3).
- (d) The Progression Mode (PMN) contains the value obtained in paragraph 3.1.3b(4).

#### 3.1.4 Input Event Notice Parameters

- a. Box.Point: This points to the terminating box.
- b. DTNN: This was the Integration Group number (DTN.No) for the terminating box.
- c. PMN: This was the Progression Mode (PMN) of the terminating box.
- d. XIT: This was the exit selected by the terminating box.

#### 3.1.5 Data Items Set or Used

- a. Own.Box Data
  - (1) (Box.Stat)g - Box status
  - (2) (CMN)a,i - Contractor manpower
  - (3) (GMN)a,i - Government manpower
  - (4) Box.Type - Type of box
  - (5) (Box.Pointer)s - From selected Successor List (YList or NList).
- b. Successor Box (S-Box) Data
  - (1) Box.Type
  - (2) DTN - Group Dependency
  - (3) PList - Predecessor List items
    - (a) (SLA)p Start Logic Assigned
    - (b) (Gp.Ctrl)p DTN.No continuity controller
    - (c) (PMA)p Progression Mode assignment
  - (4) DTN.No - Group Number
  - (5) (FPDN)g - First Pred Input Received indicator



- (6) (RPDN)g - Logical "OR" Pred Input Received indicator
- (7) (PDN)n,g - Pred Input Status indicator vector
- (8) (Pr.Wt)g - Predecessor Wait
- (9) In.Wt - Assigned Internal Wait
- (10) (CIn.Wt)g - Cumulative Internal Wait
- (11) (Box.Stat)g Box Status
- (12) (It.Ct)g Iteration Count.

c. Local Data

- (1) Own.Box - Pointer to terminating box address
- (2) PMP - Progression Mode of terminating box
- (3) PMN - Progression Mode of S-Box
- (4) PDTN - Group Number of terminating box
- (5) XIT - Successor list selection indicator
- (6) "p" - Own.Box position on PList
- (7) "n" - Own.Box location on the Predecessor Input Status vector
- (8) IWait - Duration of Internal Wait.

d. System Items

- (1) TIME.V - Current Simulation Time
- (2) (CMNA)a - Total contractor manpower use vector
- (3) (GMNA)a - Total government manpower use vector
- (4) Rsr.History - Resource Usage History file
- (5) It.Limit - Iteration count limit.

### 3.2 Box.Proc: Box Processor

#### 3.2.1. Purpose

The Box.Proc Event Routine performs all the actions needed to simulate the functions represented by the box addressed for this scheduled Event. Since these functions are somewhat dependent on the type of box being entered, both common and box-type-unique functions are processed.

### 3.2.2. Event Scheduling

A Box.Proc Event Notice is scheduled by the Flow Event Routine whenever all entry qualifications for a box have been met, and after the Internal Wait (if any) has been observed.

### 3.2.3 Actions

#### a. Override Stop

The Flow.Override field is checked to see if a Flow Stop is indicated (Flow.Override = 1). If so, the program returns with no further action. Otherwise the action continues as follows.

#### b. The following initiating Event Notice parameters are saved:

- (1) Box.Point becomes the pointer (Own.Box) to the box to be processed.
- (2) DTNN becomes the Integration Group Number (DTN.No)g for this access.
- (3) PMN is saved to indicate whether the Progression Mode is Forward or Iterative. This is needed for any subsequent Flow Event Notice created per paragraph 3.2.5.

#### c. Event Time Storage

- (1) If this is a first entry to the box ((It.Ct)g = 1) the Earliest Start Time (EST)g is set to now (TIME.V).
- (2) In all cases, the Latest Finish Time (LFT)g is set to now (TIME.V).

#### d. Box Type Processor Selection

The Box.Type field is checked so that the appropriate Box Type processing is conducted; viz.:

<u>Type</u>	<u>Function</u>	<u>Paragraph</u>
2 or 3	(Activity)	e
5	(Decision)	f
6	(Counter)	g
7	(Special Event - Milestone)	h
8 or 9	(Special Event - Other)	TBD
others	(Undefined)	

e. Activity Box Processing

(1) Override Skip Actions

If the Flow.Override field indicates that the box is to be skipped (i.e., = 2), an immediate Flow Event Notice (TIME.A = TIME.V) is scheduled per paragraph 3.2.3e(5), with XIT = 0. Otherwise:

(2) The box is put into "ON" state ((Box.Stat)g = 4).

(3) The duration of the activity (IDurnn) is computed. On the initial program this is set equal to the Assigned Duration (IDur)g,i.

(4) IDurnn is added to the (cumulative) Total Duration (ADur1)g.

(5) The Latest Finish Time (LFT)g is set to TIME.V + IDurnn.

(6) Each element of the contractor manpower use vector (CMNU)a and the government manpower use vector (GMNU)a are computed. In Model 0 these are set equal to input items (CMN)a,i and (GMN)a,i respectively.

(7) The assigned manpower use values ((GMNU)a and (GMNU)a) are added to the network totals ((CMNA)a and (GMNA)a). These are then stored (with TIME.V) in (Rsr.History).

(8) The man-days expended by the contractor (CMND)a and by the government (GMND)a are computed; i.e.:

$$(CMND)a = (CMNU)a \times IDurnn$$

$$(GMND)a = (GMNU)a \times IDurnn$$

(9) The man-day expenditures are added to the cumulative totals for the box (CME)a,g and (GME)a,g; i.e.:

$$(CME)_{a,g} = (CME)_{a,g} + (CMND)_a$$

$$(GME)_{a,g} = (GME)_{a,g} + (GMND)_a$$

- (10) (CMND)<sub>a</sub> and (GMND)<sub>a</sub> are added to the (cumulative) full system man-day totals (FNETCM)<sub>a,m</sub> and (FNETGM)<sub>a,m</sub>, respectively. This requires that the man-days be assigned to one (or more) of the monthly time slots starting with the award of the contract. If more than one month is spanned, the man-days are assigned proportionate with the relative occupancy of each included month.
- (11) If Own.Box belongs to a Subnetwork, the monthly distributed values of man-days (per Step 10) are added to the Subnetwork manpower use monthly totals (SNETCM)<sub>a,m</sub> and (SNETGM)<sub>a,m</sub>.
- (12) If this is an iterative reentry to this box ((It.Ct)<sub>g</sub> GT 1), the iterated cumulative totals are updated; i.e.:
  - (a) Add IDurrrn (duration) to (ADur2)<sub>g</sub>.
  - (b) Add (CMND)<sub>a</sub> (contractor man-days) to (CME2)<sub>a,g</sub>.
  - (c) Add (GMND)<sub>a</sub> (government man-days) to (GME2)<sub>a,g</sub>.
- (13) A Flow Event Notice is created per paragraph 3.2.5 below.
  - (a) Event Time (TIME.A) is set equal to now (TIME.V) plus the activity duration (IDurrrn).
  - (b) Box Exit (XIT) is set to zero.

f. Decision Box Processing

- (1) Override Exit Selection
  - (a) If Flow.Override = 2 ("Yes" exit) set XIT = 0.
  - (b) If Flow.Override = 3 ("No" exit) set XIT = 1.
  - (c) For the above cases, the action continues at step (3).
  - (d) Otherwise, normal processing continues at step (2).
- (2) Normal Exit Resolution
  - (a) The exit probability parameter (pYES)<sub>i</sub> is used as input to the SIMSCRIPT II.5 RANDOM function to select the exit (XIT).
  - (b) The exit selected (XIT) is saved for statistical processing as (XIT)<sub>g,i</sub>.

(3) A Flow Event Notice is created per paragraph 3.2.5.

(a) Event time (TIME.A) is set to now (TIME.V).

(b) Box exit (XIT) is set to the value obtained in step (1) or Step (2), as applicable.

(4) The Box status is set to OFF (Box.Stat)g = 0.

g. Counter Box Processing

(1) Progression Mode check - Counter Box actions are to be taken only when the box is entered during Forward Progression. If the box is addressed on Iterative Progression (PMN = 1), the Ever Routine returns without taking any further action.

(2) Override Exit selection - This is the same as for the ordinary Decision Box per paragraph 3.2.3f(1).

(3) Normal Exit Resolution

(a) The Counter Box Integration Group type attribute (DTN) establishes the Group (1 = DIG, 2 = TIG) to which the box belongs. Based on the DTN field content, the maximum Group number (DTN.Max) is set equal to CPCI attribute NDIG or NTIG. (These are both input parameters.)

(b) The Integration Group number DTNN provided within the Event Notice is then compared with DTN.Max to select a flow exit (XIT) as follows:

- If DTNN = DTN.Max, set XIT = 0("Yes" exit).
- If DTNN < DTN.Max, set XIT = 1("No" exit).
- Otherwise, produce an Error Halt giving: Box.ID, Box.Type, DTNN value, and Current Time (TIME.V).

(4) A Flow Event Notice is created per paragraph 3.2.5:

(a) Event time (TIME.A) is set to now.

(b) Box Exit (XIT) is set to the value obtained in step (2) or (3), as applicable.

(c) In the event of an Error Halt per paragraph (3)(b), no Event Notice is created.

(5) The box is put into OFF status (Box.Stat)g = 0.

h. Special Event - Milestone Processing.

(1) A Flow Event Notice is created per paragraph 3.3.5

- (a) Event time (TIME-) is set to now
- (b) Box Exit (XIT) is set to zero
- (c) Progression Mode (PMN) is set to forward.

#### 3.2.4 Input Event Notice Parameters

- a. Box.Point - Designates the box involved
- b. DTNN - Integration Group number for this event
- c. PMN - Progression Mode; Forward or Iterative.

#### 3.2.5 Output Event Notice Parameters (For a subsequent Flow Event)

- a. TIME.A - Scheduled event time
- b. Box.Point - Designates Own.Box address
- c. DTNN - Integration Group number output (always the same as Integration Group number input).
- d. PMN - Progression Mode (always the same as its value on input Event)
- e. XIT - Exit selected.

#### 3.2.6 Data Item Set Or Used

##### a. Own Box Table Data

##### (1) Common Items

- (a) (Box.Type) - Box function
- (b) (Box.Stat)g - Box status
- (c) (DTN) - Group type
- (d) (DTN.No) - Integration Group number
- (e) (It.Ct)g - Iteration count
- (f) (Flow.Override) - Override Indicator
- (g) (EST)g Earliest Start (or Occurrence) Time
- (h) (LFT)g - Latest Finish (or Occurrence) Time

##### (2) Activity Box Items

- (a) (IDur)g,i - Activity duration; input
- (b) (ADur1)g - Activity duration; cumulative total
- (c) (ADur2)g - Activity duration; cumulative; iteration only
- (d) (CMN)a,g,i - Assigned manpower; contractor; input
- (e) (GMN)a,g,i Assigned manpower; government; input
- (f) (CME1)a,g - Expended manpower; contractor; total man-days
- (g) (CME2)a,g - Expended manpower; contractor; iterated man-days
- (h) (GME1)a,g - Expended manpower; government, total man-days
- (i) (GME2)a,g - Expended manpower; government; iterated man-days.

(3) Decision Box Data

- (a) (pYES)i - Yes exit probability; input
- (b) (XIT)g,i - Exit selection, saved.

b. System Data

- (1) (TIME.V) - Current Simulation Time (now).
- (2) (NDIG) - Quantity of DIGs, this CPCI
- (3) (NTIG) - Quantity of TIGs, this CPCI.

c. Local Data

(1) Common Items

- (a) (DTNN) - DTN number
- (b) (PMN) - Progression Mode
- (c) (XIT) - Exit selected

(2) Activity Box Items

- (a) (IDurrrn) - Activity duration, computed (days)
- (b) (CMNU)a - Manpower use rate, contractor
- (c) (GMNU)a - Manpower use rate, government

(d) (CMND)a - Manpower expended, man-days, contractor

(e) (GMND)a - Manpower expended, man-days, government.

(3) Other Box Type Item

(a) DTN.Max - The maximum quantity of this group.

d. Network Data

(1) (FNETCM)a,m - Total men being used, per month, full network, contractor

(2) (FNETGM)a,m - Total men being used, per month, full network, government

(3) (SNETCM)a,m - Total men being used, per month, each Subnetwork, contractor

(4) (SNETGM)a,m - Total men being used, per month, each Subnetwork, government

(5) (Rsr.History)a - Total Men in use now; (sequential file).



## APPENDIX G

### DEMONSTRATION MODEL (M0) STATISTICS DEVELOPMENT

To produce the planned Simulator reports, the following data will be collected during each Single Simulation Pass (SPASS) and aggregated for the many SPASS repetitions to obtain statistical results. Paragraph 1 of this appendix defines the SPASS gathered data; Paragraph 2 defines the SPASS repetition data; Paragraph 3 briefly describes the content of the output report produced; and Paragraph 4 identifies the data to be entered by the user in order to conduct a Simulation run. All Integration Group (DTN) dependent data will be collected per Group (DIG or TIG), unless otherwise indicated. Data marked with an asterisk (\*) may not be obtained until the Prototype Model (M1).

#### 1. SPASS DATA COLLECTION

##### 1.1 Activity Box (A.Box) Data

The following collected for each Activity Box.

##### a. Wait Durations

- (1) Pr.Wt - Time spent waiting for the box's predecessors (Preds). It starts when the first Pred is completed, and ends when the last required Pred is completed.
- (2) CIn.Wt - The cumulative sum of all Internal Waits (In.Wts) sustained by the box. Each Internal Wait period reflects a delay in starting that is inherent in the function. Each In.Wt begins when all Pred input requirements are satisfied and ends when the specified Internal Wait period completes. This only applies to boxes on which In.Wt is not zero. The full value of In.Wt is applied to each DTN and to each iteration.
- (3) \*RWt1 - The sum of all Waits that occur when an activity is otherwise able to begin, but cannot because manpower or other needed resources are not available (e.g., the development or test facilities are currently fully used, or are "down").
- (4) \*RWt2 - Same as RWt1, except that only the Waits associated with iterative reentry are included.

##### b. Activity Duration

- (1) Total (ADur1) - The time during which an activity is in actual operation. It starts immediately after all Waits have completed and continues until the activity is ended. Each time the A.Box

is reentered for iteration, the incremental duration is added to the accumulated duration.

- (2) Iterative (ADur2) - This is the same as ADur1 except that only the iterative reentry durations are accumulated.

c. Event Times

- (1) Earliest Start Time (EST) - The earliest time at which the function could begin, assuming no constraints other than completion of all required predecessors.
- (2) Latest Finish Time (LFT) - The latest time at which the activity ends. If the Activity Box is entered several times (e.g., for iteration) LFT will retain the simulated time when the last activation ends.

d. Manpower Use Data

- (1) (CME1)a - The total contractor manpower used in man-days per manpower type is collected for each Activity Box. This includes the sum of both the first use and all iterative reentry uses.
- (2) (CME2)a - This is the same as (CME1)a except that only the iterative reentry use is included.
- (3) (GME1)a - The same as (CME1)a except that it applies to each government manpower type.
- (4) (GME2)a - Same as (GME1)a, except that only iterative reentry use is included.

e. Iteration Data

- (1) It.Ct - The number of times that the box was activated (first operation plus each subsequent iterative operation).

1.2 Decision (Branch) Box (B.Box) Data

The following information is collected for each branching Decision Box.

a. Wait Duration

- (1) Pr.Wt - Same as paragraph 1.1a(1).
- (2) CIn.Wt - Same as paragraph 1.1a(2).

b. Event Times

- (1) EST - Same as paragraph 1.1c(1).

- (2) LFT - The time of last entry into this Decision Box; e.g., as a result of iteration. If the box is entered only once, LFT = EST.

c. Iteration Data

- (1) It.Ct - Same as paragraph 1.1e(1)
- (2) (XIT)i - The exit selected ("Yes" or "No") on each iteration (contains room for up to "n" binary choices, where "n" is the maximum number of allowable iterations).

1.3 Counter Box (C.Box) Data

The following information will be collected for each counter Decision Box.

EST - Each time the C.Box is entered the Simulation Time (TIME.V) is retained. Same as EST, paragraph 1.2b(1).

1.4 Subprocess Box Data

In Model 0 a Subprocess Box is the same as Activity Box. The box distinction is intended for possible future use.

1.5 Support (Helper) Box Data

This is the same as an Activity Box. The distinction is intended for future use.

1.6 Special Event Box (E.Box) Data

The following information will be collected for each Special Event Box.

a. Milestone Type

- (1) EST - Records the Simulation Time (TIME.V) of each entry. Same as EST, paragraph 1.2b(1). (Milestone Special Event Boxes should not be entered via iteration).

b. Other Types (TBD).

1.7 Subnetwork Data

The following information will be collected for each of the 15 possible Subnetworks definable by the user.

a. Manning and Manpower Use - This will aggregate the total manning and manpower use by all activities within the network, as follows:

- (1) (CSNETMM)a - The contractor manpower usage (in man-days) on each Subnetwork for each manpower type accumulated for each month

(i.e., aggregates of 20 working days) from start of contract. Values for any activities which span month boundaries will be proportionately subdivided among all such months.

- (2) (GSNETMM)a - Same as (CSNETMM)a, except that they cover government manpower types.

b. \*Other Resource Utilization - The total utilization of each defined support resource (e.g., programming and test facilities) will be accumulated on a monthly basis.

### 1.8 Full Network Data

a. The data defined for Subnetworks (see paragraph 1.7) will be collected for the entire network.

b. (CMNA)a - A list of all contractor personnel manpower usage transactions will be collected.

c. (GMNA)a - Same as (CMNA)a, except for government personnel.

## 2. PASS REPETITION DATA

At the conclusion of each SPASS the box-related data retained per paragraphs 1.1 - 1.8 above will be aggregated to obtain statistical results, as follows:

### 2.1 General Box-related Data

Each of the SPASS data items identified in the following table will be:

- a. summed to obtain its total; and  
b. squared and summed to obtain the basis for its variance:

<u>Par. #</u>	<u>Element</u>	<u>Data Name</u>	<u>Definition</u>
1.1a(1)	A.Box	Pr.Wt	Predecessor Wait Time
(2)	"	CIn.Wt	Process Wait Time
(3)	"	*RWt1	Resource Wait Time (Total)
(4)	"	*RWt2	Resource Wait Time (Iterative)
1.1b(1)	"	ADur1	Total Activity Duration
(2)	"	ADur2	Iterated Activity Duration
1.1c(1)	"	EST	Earliest Start Time
(2)	"	LFT	Latest Finish Time
1.1d(1)	"	(CME1)a	Total Contractor Manpower Usage
(2)	"	(CME2)a	Iterated Contractor Manpower Usage
(3)	"	(GME1)a	Total Gov't Manpower Usage

<u>Par. #</u>	<u>Element</u>	<u>Data Name</u>	<u>Definition</u>
(4)	"	(GME2)a	Iterated Gov't Manpower Usage
1.1e(1)	"	It.Ct	Iteration Count
1.2a(1)	B.Box	Pr.Wt	Predecessor Wait Time
(2)	"	CIn.Wt	Decision Wait Time
1.2b(1)	"	EST	First Time of Decision
(2)	"	LFT	Last Time of Decision
1.2c(1)	"	It.Ct	Iteration Count
1.3	C.Box	EST	Count Increment Time
1.6a	E.Box	EST	Time of Event
1.7a(1)	SUBNET	(CSNETMM)a	Contractor Manpower
(2)	SUBNET	(GSNETMM)a	Government Manpower
1.8a(1)	FULNET		Contractor Manpower
(2)	FULNET		Government Manpower

In addition, a count of the number of times each box was entered for the first or only time (i.e., ignoring iterative reentry) will be maintained.

## 2.2 Decision Box Exit Summaries

SPASS data will be processed to extract exit history for each B.Box as a function of the iteration number, based on (XIT)i data per paragraph 1.2c(2); e.g.:

Iteration No. (It.Ct)	1	2	3	4	5
Quantity of Exits	100	69	35	6	0
"Yes" Exit Probability	.31	.49	.82	1.00	

## 2.3 Subnetwork and Full Network Data

These data will be extracted as follows:

### a. Timing Data

The Earliest Start Time (EST) and Latest Finish Time (LFT) (per Subnetwork) obtained in each SPASS will be summed to obtain each total, and squared and summed to obtain each variance. Also, the most extreme values (earliest and latest) of each will be retained.

**b. Manpower Data**

- (1) The contractor manpower expended (in man-months per month, (CSNETMM)a) will be summed, and squared and summed to obtain the total and the variance.
- (2) The government manpower expended (GSNETMM)a will also be treated as in (1) above.
- (3) Corresponding manpower level statistics will be developed from (CFNETMM)a and (GFNETMM)a.
- (4) Statistics on total manpower expended (in man-months) for each type of contractor and government manpower will be obtained from the corresponding monthly totals for the whole contract period.
- (5) \*A profile of daily manpower use (by type) will be obtained by converting CMNA and GMNA transactions into daily totals.

**c. \*Dollar Cost Data**

- (1) Manpower dollar cost data will be derived for each manpower category listed in paragraph 2.3b above, by the application of entered labor rate values. Overhead and General & Administrative rates will also be reflected.
- (2) \*Other resource dollar costs will be treated in later versions.
- (3) Total monthly, annual, and full project dollar costs will also be developed.

**3. REPORTS**

Reports that include means and standard deviations, based on the accumulations defined above, the corresponding box entry counts, and appropriate box, Subnetwork, and full network identification will be prepared at the end of each Simulator run (i.e., after all repetitions). A simple project schedule that includes defined milestones will also be prepared. These reports' formats are to be determined.

**4. SIMULATOR INPUT DATA**

The Simulator Model 0, which will be used for project cost and schedule forecasting, will require the following data to be entered for each CPCI being simulated.

- a. The network linkage definition per Table B-1;
- b. Override information which will customize the network configuration to reflect the specific procurement condition; (This would be accomplished by the

addition and deletion of function boxes and by the corresponding modification of the Predecessor and Successor lists for the remaining function boxes.)

c. Parameter data per Tables B-2 through B-4 that is customized to include any new boxes added per step b and by value changes that reflect the specific procurement (including manpower category changes, if applicable);

d. The quantity of DIGs planned for the development, and the proportion (of the effort) consumed on each DIG;

e. The quantity of TIGs planned for the qualification testing and the proportion of effort consumed on each TIG; and

f. Miscellaneous project data that includes the project starting date, its estimated duration, and the number of simulation repetitions to be conducted.

APPENDIX H

DATA REPORTING FOR  
LATER SIMULATOR MODELS

1. STRUCTURES OF INTEREST

Following are several Process Model structures about which users may want information, and for which the Process Model simulation program (i.e., the Simulator) must therefore collect statistics. Activity Boxes, Decision Boxes and Special Event boxes are the basic elements from which the others are composed. Subprocess Calls are each shorthand notation for a group of these basic elements. Paths, Subnetworks, and the entire process are respectively higher-level aggregates of basic elements and Subprocess Calls. Activity Boxes, Decision Boxes, Special Event Boxes and Subnetworks have been explained elsewhere in this report. Subprocesses and Paths are explained below.

A Subprocess is any portion of the Model logic that (1) comprises a connected set of two or more basic elements or Subprocess Calls (to the same or to another Subprocess); and that (2) has a single entrance and a single exit. A Subprocess Call is any element of the Model logic that stands for an instance of a specified Subprocess. A Subprocess Call is exactly like a closed subroutine call in a programming language. Like a subroutine call, a Subprocess Call may have input or output parameters.

A Path is a specific single-thread sequence of Activity Boxes, Decision Boxes, and Subprocess Calls. A Path can be defined by a list of connected Activity Boxes, Subprocess Call boxes, and specific Decision Box exits. A Path may include no concurrent activities and may include only a single exit from each of its Decision Boxes. A Path may include a loop. Examples of Paths (see Figure A-2) include:

- a. 2A, 6F, 6G, 6H, 6I(Y)
- b. 6G, 6H, 6I(N).

Example b. includes a loop.

Path statistics may help Simulator debugging and may also help reveal elapsed times and costs along single-thread sequences. Ability to trace a Path may be important, too.

Paths are special cases of Subnetworks. No special Path statistics will be provided, since a Subnetwork can be defined to represent any Path.

2. SPECIFIC REPORT TYPES

The information contents of the reports desired from Simulator Models 1, 2, & 3 are outlined below, as are the input data needed to specify them.



These reports' exact formats have yet to be defined. Asterisks distinguish the different Models' report contents. Those that Model 1 should provide are not flagged. The additional reports or report components that are candidates for Model 2 are marked by a single asterisk (\*), while those that Model 3 or later will provide are flagged by two asterisks (\*\*). The same conventions apply to the inputs needed. Most of the reports identified below are tabular. Graphic equivalents of most of these will be provided, some as early as Model 1. Superimposed plots of selected simulation results, of the same or different type (if compatible) from the same or successive simulation runs will also be available.

## 2.1 For the Overall Process

### 2.1.1 General

#### 2.1.1.1 Inputs Needed

##### a. Number of Simulator Repetitions

Note: This is the number of times that Simulator execution will repeat, with the same inputs but with different pseudo-random number seeds, to smoothe out random errors. In Model 2 or later we hope to include an adaptive algorithm that will check for convergence as a function of defined statistical confidence limits, and that will stop repetition sooner if and when simulation results within the limits have been achieved.

##### b. Program Start Date (Month, Day & Year)

##### c. System Structure Indicators

- (1) Number of System Segments\*
- (2) Number of CPCIs
- (3) Number of Design Integration Groups (DIGs) per CPCI
- (4) Number of Test Integration Groups (TIGs) per CPCI
- (5) Number of CPCs per CPCI
- (6) Number of new CPCs per DIG
- (7) Intra-CPCI Dependencies at the DIG Level.

Note: E.g., that DIG-II of CPCI 1 may not begin until DIG-I of CPCI 3 is finished.

##### d. System Definition Matrix\*\*

Note: The system definition matrix will replace the system structure indicators (input c.) in Simulator Model 2 and later. The Simulator

use of the system definition data (e.g., in parametric equations) will explicitly reflect differences in computer program size, difficulty, development approach, etc.

(1) Rows: System Components

Note: The system itself, each of its segments, each of its segment's CPCIs, and each CPI's CPCs are the system components.

(2) Columns: System Component Identifier; DIG Number of First Development; First DIG Development Priority (1 = High); Test Priority (1 = High); SARE Parameter Data Types; Dependencies.

Note: The system component identifier specifies each system component's level and hierarchical position. The DIG number of first development specifies when that system component will first be started. The first TIG number specifies when the system component will first be tested. The SARE parameters specify each system component's presumed size, difficulty, etc.

e. Manpower Availability Matrix\*

(1) Rows: Months Since Program Start

(2) Columns: Contractor & Government Labor Categories

(3) Elements: Number of Persons Available

Note: Some future Simulator version may want to represent manpower that is somewhat interchangeable by type (as people really are). This may entail representing numbers of people who have several skills, and assigning to a task any person who has such a skill (subject, perhaps, to assignment priorities). This outline does not reflect this concept further.

f. Other Resource Availability Matrix\*

(1) Rows: Months Since Program Start

(2) Columns: Other Resource Types

(3) Each Element:

(a) Other Resource Quantity

(b) Quantity's Unit of Measure.

g. Integration Tree Matrix\*\*

(1) Rows: System Components & Other Integrands

Note: "Other Integrands" = modules and interim results of integration that are not system components.

- (2) Columns: TIGs
- (3) Each Element:
  - (a) Component ID of Integration Result
  - (b) Component IDs of All Integrands.

h. Simulation Report Selection Matrix\*\*

- (1) Rows: The Entire System; All Segments; Each Segment; All CPCI's; Each CPCI; All CPCs; Each CPC
- (2) Columns: System Component Level; System Component Name; The Report Types Appropriate to Each Row
- (3) Each Element: Level Number; System Component Name (or "All"); Whether to Produce the Report

Note: Many of the reports identified below will output data for each of the acquisition program's calendar months during which manpower or other resources are spent. Simulator Model 1 will assume a fixed number of working days (i.e., 20) per calendar month. Subsequent more sophisticated versions (i.e., Model 2 or later) may develop results per a user-specified, variable number of working days per month. If so, the Model input should include the following vector:

i. Program Month Composition Vector\*\*

- (1) Rows: Calendar Months Since Program Start
- (2) Each Element: Number of Working Days in this Month (or Number of Hours Worked / 8) (Precision = .1 day).

2.1.1.2 Output Produced

- a. Listing of All Simulation Program Inputs as Entered
- b. See Sections 2.1.2 - 2.1.7 below.

2.1.2 Manpower Use

2.1.2.1 Additional Inputs Needed

- a. n1 = Number of Standard Deviations (Sigma) for Optimistic Estimates
- b. n2 = Number of Standard Deviations (Sigma) for Pessimistic Estimates

#### 2.1.2.2 Output Produced

a. Mean (Expected Value) Manpower Expenditure Matrix

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals

Note: The Simulator will be designed so that the numbers of contractor and government manpower types, as well as their identities, can easily be changed. Need for such alteration is likely as the Simulator's application changes.

- (3) Each Element: Mean Number of Man-Days Expended (in All Repetitions).

b. Manpower Expenditure Variance Matrix

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: Variance of Number of Man-Days Spent

Note: We currently foresee four sources of manpower variation: (1) different numbers of initial activity executions due to random decision box alternative selection; (2) different numbers of iterations due to random loop control decision outcomes; (3) in Model 3 and later, different manpower expenditure per activity and per activity iteration due to random sampling of manpower level and duration based 3-value (PERT-like) estimates; and (4) differences in monthly values due to the different delays that result from the above factors and different manpower constraints.

c. Optimistic Manpower Use Matrix

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: (Mean - n1 Sigma) Number of Man-Days Spent.

d. Pessimistic Manpower Use Matrix

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: (Mean + n2 Sigma) Number of Man-Days Spent.

Note: Corresponding reports showing manning levels (vice man-days) will also be provided.

**2.1.3 Other Resources Use\***

**2.1.3.1 Additional Inputs Needed\***

- a. n3 = Number of Standard Deviations (Sigma) for Optimistic Estimates\*
- b. n4 = Number of Standard Deviations (Sigma) for Pessimistic Estimates\*.

**2.1.3.2 Output Produced\***

- a. Mean (Expected Value) Other Resource Use Matrix
  - (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
  - (2) Columns: Resource Types and Units of Measure
  - (3) Each Element: Expected Quantity of this Resource Used.
- b. Other Resource Use Variance Matrix
  - (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
  - (2) Columns: Resource Types and Units of Measure
  - (3) Each Element:
    - (a) Number of Initial Activity Executions Entailing Use of this Resource Type During this Month.
    - (b) Variance of Quantity of this Resource Used.
- c. Optimistic Other Resource Use Matrix
  - (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
  - (2) Columns: Resource Types and Units of Measure
  - (3) Each Element: (Mean - n3 Sigma) Quantity of Each Resource Used.
- d. Pessimistic Other Resource Use Matrix
  - (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
  - (2) Columns: Resource Types and Units of Measure
  - (3) Each Element: (Mean + n4 Sigma) Quantity of Each Resource Used.

#### 2.1.4 Dollar Cost

##### 2.1.4.1 Additional Inputs Needed

###### a. Assumed Salary + Overhead Rate Matrix

- (1) Rows: Months Since Program Start
- (2) Columns: Contractor and Government Labor Categories
- (3) Elements: Undiscounted Hourly Labor Rate (Including Overhead) for Each Month Since Program Start

Note: The proposed reports do not distinguish between regular and overtime pay. To do so the Simulator would need to handle separate rates and manhour estimates for both.

###### b. Assumed Other Resource Rate Matrix\*

- (1) Rows: Months Since Program Start
- (2) Columns: Resource Types
- (3) Each Element: Undiscounted Rate per Unit of this Resource Type

Note: These units must be the same as those specified for in the other Resource Availability Matrix (see paragraph 2.1.1.f).

Note: The Simulator (Model 2 and later) should calculate costs as of the valuation date (input c.). Using the Assumed Interest Rate Vector (input d.), the costs of work performed before the Valuation Date should be accumulated (i.e., increased) and the costs of work to be performed after the valuation date should be discounted (i.e., decreased).

###### c. Valuation Date (Month & Year).

###### d. Assumed Interest Rate Vector

Note: The Simulator provides for input of an interest rate vector to reflect the effect of anticipated changes in interest rate. However, the Simulator should optionally accept a single rate and create a (constant) vector from it.

Note: To develop costs that ignore interest, the user should input no Assumed Interest Rate Vector.

- (1) Rows: Months Since Program Start
- (2) Each Element: Assumed Annual Simple Interest Rate Applicable to the Month.

#### **2.1.4.2 Output Produced**

**a. Mean (Expected Value) Manpower Cost Matrix**

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of Mean Manpower of this Type Expended.

**b. Optimistic Manpower Cost Matrix**

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of (Mean -  $n1$  Sigma) Manpower of this Type Expended.

**c. Pessimistic Manpower Cost Matrix**

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Manpower Types; Contractor, Government, & Grand Totals
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of (Mean +  $n2$  Sigma) Manpower of this Type Expended.

**d. Mean (Expected Value) Other Resource Cost Matrix\***

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Resource Types; Government & Contractor Totals; Grand Total
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of Mean Other Resource Use.

**e. Optimistic Other Resource Cost Matrix\***

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Resource Types; Government & Contractor Totals; Grand Total
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of (Mean -  $n3$  Sigma) Other Resource Use.

**f. Pessimistic Other Resource Cost Matrix\***

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Resource Types; Government & Contractor Totals; Grand Total
- (3) Each Element: Accumulated or Discounted Dollar Value as of Valuation Date of (Mean + n4 Sigma) Other Resource Use.

g. Mean (Expected Value) Total Dollar Cost Matrix\*

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Contractor & Government Totals; Grand Total
- (3) Each Element: Sum of the Corresponding Column from the Mean Manpower Cost and Other Resource Cost Matrices.

h. Optimistic Total Dollar Cost Matrix\*

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Contractor & Government Totals; Grand Total
- (3) Each Element: Sum of the Corresponding Columns from the Optimistic Manpower Cost and Other Resource Cost Matrices.

i. Pessimistic Total Dollar Cost Matrix\*

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns: Contractor & Government Totals; Grand Total
- (3) Elements: Sum of the Corresponding Columns from the Pessimistic Manpower Cost and the Optimistic Other Resource Cost Matrices.

2.1.5 Elapsed Time

2.1.5.1 Additional Inputs Needed

a. Gantt Chart Content & Arrangement Option(s)

- (1) Chart Content Options
  - (a) Milestones (& Corresponding Box IDs) Only
  - (b) Milestones & Subnetworks Only
    - (i) All Milestones & Subnetworks
    - (ii) List-specified Subset Only
  - (c) Milestones Plus Activity & Decision Boxes Only\*



- (i) All Activities & Decisions
- (ii) List-specified Subset Only
- (d) Milestones, Subnetworks, & Component Activities & Decisions\*
  - (i) All Milestones, Subnetworks, & Component Activities & Decisions
  - (ii) List-specified Subset Only

(2) Line Content Options

- (a) Mean Initial Start Time to Finish Time
- (b) Four Components\*
  - (i) Wait for Immediate Predecessors
  - (ii) Internal Wait Time
  - (iii) Wait for Resources
  - (iv) Activity Duration

(3) Arrangement Options

- (a) Network Definition Table Order
- (b) By Subnetwork Number
- (c) By Increasing Mean Start Time (at Each Level Included)\*
- (d) User-Specified Order\*.

- b. n5 = Number of Standard Deviations for Optimistic Estimates
- c. n6 = Number of Standard Deviations for Pessimistic Estimates

2.1.5.2 Output Produced

- a. Gantt Chart of Milestones

Note: For Model 1 this may only be a list of Milestones and associated dates.

- b. Gantt Chart of Milestones Plus Mean Start & Finish Times.
- c. Gantt Chart of Milestones Plus Optimistic Start & Finish Times.
- d. Gantt Chart of Milestones Plus Pessimistic Start & Finish Times.

e. Gantt Chart of Milestones Plus Optimistic, Mean, & Pessimistic Start & Finish Times.\*

f. Network Logic Diagram Showing Milestones Plus Boxes Annotated by Optimistic, Mean, & Pessimistic Start & Finish Times.\*\* The diagram will highlight the critical path through the network.

#### 2.1.6 Resource Constraint History\*

##### 2.1.6.1 Additional Inputs Needed\*

- a.  $n7$  = First Sampling Day
- b.  $n8$  = Sampling Interval

Note: The Resource Constraint History Tables (See paragraph 2.1.6.2a) present values of resources provided, in use, in demand but unavailable, etc., as of a specified sample of days since acquisition program start.  $n7$  and  $n8$  (both integers) specify the first such day and the number of days between successive samples.

##### 2.1.6.2 Output Produced\*

a. Resource Constraint History Tables (One per Manpower or Other Resource Type)

###### (1) Once-per-Table Data

- (a) Resource Type Identifier
- (b,c) Maximum & Minimum Value of Resource Provided ( $R1$ )
- (d,e) Maximum & Minimum Value of Resource In Use ( $R2$ )
- (f,g) Maximum & Minimum Value of Resource Unused ( $R3$ )
- (h,i) Maximum & Minimum Value of Unsatisfied Demand ( $R4$ )
- (j,k) Maximum and Minimum Number of Processes in Execution ( $Ne$ )
- (l,m) Maximum & Minimum Number of Processes Delayed because of this Resource was Unavailable ( $Nd$ )

###### (2) Time History of Resource Status

- (a) Rows: Selected Days Since Program Start (i.e.,  $n7$ ,  $n7 + n8$ ,  $n7 + 2n8$ , etc.)
- (b) Columns:
  - (i) Column 1: Day Number of Sample

(ii) Columns 2 - 8: R1, R2, R3, R4, R2 + R3 + R4; Ne; Nd

(iii) Each Element: Value of the Variable at Start of the Day

Note: In Model 2 or later, these tables should be supplemented by plotted charts.

b. Resource Shortage Impact Matrices (One per Manpower or Other Resource Type)

(1) Rows: Calendar Months Since Program Start; FY & Grand Totals

(2) Columns:

Note: Each column presents a type of value related to the availability of this resource. Each element shows the value of this type for the time period represented by the row. The elements of columns 4, 6, 7, and 8 quantify time-related losses during each such time period that happen because (at least in part) a shortage of this resource prevents one or more processes from starting or beginning iteration during the time period.

Note: A process may need more than one type of resource, and may be delayed because too little of two or more such resource types are available. In such cases the Simulator will collect unavailability data independently for each such resource, during whatever periods the resource is in short supply.

(a) Column 1: Number of Resource Unit-Days (e.g., Man-Days) Provided

(b) Column 2: Number of Resource Unit-Days Used

(c) Column 3: Number of Available Resource Unit-Days Unused

(d) Column 4: Number of Unavailable Resource Unit-Days

Note: This is the sum, for each process delayed because it couldn't get enough of the resource, of each required amount of the resource times the number of days the process was delayed because it couldn't get the needed amount.

(e) Column 5: Sum of Column 2, 3, & 4 Values

Note: This may often somewhat exceed the Column 1 value.

(f) Column 6: Number of Different Processes Delayed

(g) Column 7: Number of Process-Days Lost

Note: This is the sum of the delays incurred during the period by all processes delayed (during the period) because (in part at least) they couldn't get enough of the resource.

(h) Column 8: Normalized Size of the Shortage

Note: Column 4/Column 7. This is a rough estimate of the level of this resource needed over the entire period to alleviate the shortage. However, users should note that providing this level may not entirely relieve the problem, in part because of interactions among this and other constraints. E.g., relieving another bottleneck could increase the demand on this resource.

c. Resource Demand Lists\*

- (1) For any Manpower or Other Resource Type; and for any month, any fiscal year, or the entire acquisition program duration, a list of the processes consuming the resource, and the amount consumed during the period, may be requested.
- (2) A similar list of the processes unable to start or repeat during the period because they cannot obtain enough of the resource may be requested. Both lists will be arranged in decreasing order of Resource Unit-Days. These lists should help identify prime "cost-drivers".

d. Predecessor Completion Impact Matrix

- (1) Rows: Calendar Months Since Program Start; FY & Grand Totals
- (2) Columns:
  - (a) Column 1: Number of Different Processes Delayed Waiting for Essential Immediate Predecessors to Finish
  - (b) Column 2: Number of Process-Days Lost Thereby
  - (c) Column 3: Mean Number of Process-Days Lost (i.e., Column 2/Column 1)

Note: This report is intended to indicate:

- (1) limits on the benefits of overlapping concurrent activities (for a given process);
- (2) the potential for improvement by process revisions aimed at better matching the durations of concurrent process paths.

### 2.1.7 Trace

SIMSCRIPT II.5 provides a listing of each Event as it is removed from the Event queue. It also provides a program flow trace (from the main program) when a program aborts. Both of these look useful.

## 2.2 For Each Subnetwork

### 2.2.1 Additional Inputs Needed

- a. Subnetwork Membership of Each Basic Element & Subprocess Call.
- b. Same as Overall System Inputs (see paragraph 2.1.1).

### 2.2.2 Output Produced

Same as Overall System Results (see paragraph 2.1.2).

## 2.3 For Each Path

No output will be provided for Paths per se, since a user can designate each Path of interest as a Subnetwork.

## 2.4 For Each Activity Box

### 2.4.1 Additional Inputs Needed

- a. Activity Box ID.
- b. Subnetwork Number (0 = None).
- c. Subprocess ID (Blank = None).
- d. Activity Priority (1 = High).\*
- e. Responsible Agency Code (i.e., "Doer").
- f. System Level Code.
- g. Number of Immediate Predecessors.
- h. Immediate Predecessor Matrix (Same as paragraph 2.4.2a(1)(i)).
- i. Immediate Successor Vector

Each Element: Box ID of an Immediate Successor

- j. Internal Wait Time

(1) Mean Value

(2) Variance.

k. Required Manpower Matrix (For Each Integration Group)

- (1) Columns = Manpower Types; e.g., Contractor Sys, Prg, Tst, Spt; Government participating organizations.

Note: These types should be checked vs. those designed for SARE.

- (2) Rows: One per Iteration
- (3) Each Element: Number of Persons of This Type Needed per Iteration.

l. Required Other Resource Matrix (One per Integration Group)\*

- (1) Columns: Other Resource Types & Units
- (2) Rows: One per Iteration
- (3) Each Element: Number of Units of this Resource Needed for this Iteration.

m. Required Duration Vector (For each Integration Group)

- (1) One Element per Iteration
- (2) Each Element a Pair
  - (a) Expected Value of Duration
  - (b) Variance of Duration.

n. Duration Iteration Vector

- (1) One Element per Iteration
- (2) Each Element the Proportion of the Original Duration Estimate Required to Complete Each Iteration.

2.4.2 Output Produced

a. Activity Box Activation Tables (One Set per Activity Box. One Instance per Integration Group, & Total).

- (1) Once per Table Data:
  - (a) Activity Box ID
  - (b) Subnetwork Number (0 = None)
  - (c) Containing Subprocess ID (Blank = None)

- (d) Activity Priority (1 = High)\*
- (e) Integration Group Number; "Total"
- (f) Responsible Agency Code (i.e., "Doer")
- (g) System Level Code
- (h) Number of Immediate Predecessors
- (i) Immediate Predecessor Matrix
  - (i) Rows: One per Immediate Predecessor
  - (ii) Columns: Box ID (including Decision Box Exit ID); Box Type; Start Logic (see Figure A-1); Progression Mode; Group Number Controller; Iteration Loop Return? (Yes, No); Integration Group Repetition Return? (Yes, No).
- (j) Number of Immediate Predecessors Requiring Initial Wait
- (k) Mean Initial Wait for Required Immediate Predecessors
- (l) Variance of Initial Wait
- (m) Mean Initial Internal Wait Time
- (n) Initial Internal Wait Time Variance
- (o) Number of Different Manpower & Other Resource Types Needed
- (p) Mean Wait for Available Manpower & Other Resources
- (q) Variance of Wait for Available Resources
- (r) Mean Initial Activity Start Time
- (s) Variance of Initial Activity Start Time
- (t) Mean Total Activity Duration: All Iterations
- (u) Variance of Total Activity Duration: All Iterations
- (v) Mean Total Elapsed Time ((m)+(p)+(t))
- (w) Variance of Total Elapsed Time
- (x) Mean Activity Finish Time ((r)+(v))
- (y) Variance of Activity Finish Time

(z) Activity Manpower Use Vector

One Element per Manpower Type. Each Element: Total Manpower of the Type Used.

(aa) Activity Manpower Use Variance Vector

One Element per Manpower Type. Each Element: Variance of Total Manpower of this Type Used

(ab) Other Resource Use Vector\*

One Element per Other Resource Type. Each Element: Total of the Other Resource Type Used; Units

(ac) Other Resource Use Variance Vector\*

One Element per Other Resource Type. Each Element: Variance of the Total Other Resource Type Used

(ad) Mean Number of Iterations

(ae) Variance of Number of Iterations

(2) Per Iteration Data

Note: Each time a box is activated, per Simulator repetition and per Integration Group, is an iteration.

(a) Rows:

(i) Number of Activations, this Iteration (for All Simulator Repetitions)

(ii) Relative Frequency of Activation, this Iteration (i.e., Column (i) Value / Actual Number of Simulator Repetitions)

(iii) Mean Activity Duration, this Iteration

(iv) Activity Duration Variance, this Iteration

(v) Mean Activity Manpower Use Vector, this Iteration (One Element per Manpower Type)

(vi) Activity Manpower Variance Vector

(vii) Mean Activity Other Resource Use Vector, this Iteration\* (One Element per Other Resource Type)

(viii) Activity Other Resource Use Variance Vector



(b) Columns: One per Iteration.

## 2.5 For Each Decision Box

### 2.5.1 Additional Inputs Needed

- a. Decision Box ID.
- b. Subnetwork Number (0 = None).
- c. Subprocess ID\* (Blank = None).
- d. Responsible Agency Code (i.e., "Doer").
- e. System Level Code.
- f. Number of Immediate Predecessors.
- g. Immediate Predecessor Matrix Same as paragraph 2.4.2a(1)(i).
- h. Possible Decision Box Outcome Matrix
  - (1) Rows: Possible Branches
  - (2) Columns:
    - (a) Column 1: Branch ID
    - (b) Column 2: Expected Branch Probability, First Iteration
    - (c) Columns 3-n (n less than 11): Expected Branch Probability, Subsequent Iterations.
- i. Immediate Successor Matrix
  - (a) Rows: Possible Branches
  - (b) Columns: Possible Immediate Successors of Each Branch
  - (c) Each Element: Branch ID; Immediate Successor ID.

### 2.5.2 Output Produced

- a. Decision Box Activation Table (One Set per Decision Box. One Instance per Integration group, & Total).
  - (1) Once per Table Data:
    - (a) Decision Box ID
    - (b) Subnetwork Number (0 = None)

- (c) Containing Subprocess ID\* (Blank = None)
- (d) Integration Group Number; "Total"
- (e) Responsible Agency Code (i.e., "Doer")
- (f) System Level Code
- (g) Number of Immediate Predecessors
- (h) Immediate Predecessor Matrix (Same as paragraph 2.4.2a(1))
- (j) Number of Initial Predecessors Requiring Initial Wait
- (k) Mean Initial Wait for Required Immediate Predecessors
- (l) Variance of Initial Wait
- (m) Mean Time of First Occurrence
- (n) Variance: Time of First Occurrence
- (o) Mean Number of Iterations
- (p) Variance of Number of Iterations.

(2) Per Iteration Data

Note: Each time a box is activated, per simulation program repetition and integration group is an iteration.

- (a) Rows:
  - (i) Number of Activations, this Iteration (for All Simulation Program Repetitions)
  - (ii) Relative Frequency of Activation, this Iteration (i.e., Column (i) Value / Actual Number of Simulator Repetitions)
  - (iii) ID of First Branch
  - (iv) Expected Probability of Selection (p(E)): First Branch
  - (v) Relative Frequency of Selection (p(A)): First Branch
  - (vi) Variance of Relative Selection Frequency: First Branch
  - (vii) ID, p(E), p(A), & Variance\*: Second Branch

(viii) Corresponding Values for any Third, Fourth, etc.,  
Branches

(b) Columns: One per Iteration

Note: These per iteration data should help the user decide how closely the Monte Carlo simulation of branch selection follows the probabilities expected.

## **2.6 For Each Special Event Box**

### **2.6.1 Additional Inputs Needed**

- a. Special Event Box ID.
- b. Subnetwork Number (0 = None).
- c. Subprocess ID (Blank = None).
- d. Type of Special Event (1 = Milestone; 2 = Random Process Modifier).
- e. System Level Code.
- f. Number of Immediate Predecessors.
- g. Immediate Predecessor Matrix (Same as paragraph 2.4.2a(1)(i)).
- h. Immediate Successor Vector

Each Element: Box ID of an Immediate Successor.

- i. Boxes Altered Vector
- j. Parameter Vector

### **2.6.2 Output Produced**

a. Special Box Activation Table (One Set per Activity Box. One Instance per Integration Group & Total).

(1) Once per Table Data:

- (a) Special Event Box ID
- (b) Subnetwork Number (0 = None)
- (c) Containing Subprocess ID\* (Blank = None)
- (d) Special Event Type (1 = Milestone; 2 = Process Modifier)
- (e) Integration Group Number; "Total"

- (f) System Level Code
- (g) Number of Immediate Predecessors
- (h) Immediate Predecessor Matrix (Same as paragraph 2.4.2a(1)(i))
- (i) Number of Immediate Predecessors Requiring Initial Waits
- (j) Mean Initial Wait for Required Immediate Predecessors
- (k) Variance of Initial Wait
- (l) Mean Initial Special Event Start Time
- (m) Variance of Initial Activity Start Time\*
- (n) Mean Number of Iterations
- (o) Variance of Number of Iterations

(2) Per Iteration Data

Note: Each time a box is activated, per simulation program repetition and per Integration Group, is an iteration.

(a) Rows:

- (i) Number of Activations, this Iteration (for All Simulation Program Repetitions)
- (ii) Relative Frequency of Activation, this Iteration (i.e., Column (i) Value / Actual Number of Simulation Program Repetitions)

(b) Columns: One per Iteration.

**2.7 For Each Subprocess Call\***

**2.7.1 Inputs Needed\***

Same as Outputs Produced items 2.7.2a.(1)(a)-(d), etc.

**2.7.2 Output Produced\***

a. Subprocess Call Activation Table (One Set per Subprocess Call. One Instance per Integration Group & Total).

(1) Once per Table Data

(a) Subprocess Call ID

- (b) Subnetwork Number (0 = None)
- (c) Containing Subprocess ID (Blank = None)
- (d) ID of Called Subprocess
- (e) Priority of this Subprocess Call (1 = High)
- (f) Subprocess Call Input Parameter Vector

Each Element: Input Parameter ID

- (g) Subprocess Call Output Parameter Vector

Each Element: Output Parameter ID

- (h) Same as Activity Box Results Required Items (1)(e)-(ae)

Note: These results are to be developed as if the called Subprocess were a single activity.

## (2) Per Iteration Data\*

Same as Activity Box Per Iteration Data (2.4.2a.(2)).

## 2.8 For Each Subprocess\*

Note: Each component of a subprocess is presumed defined as a basic element or subprocess call. See paragraphs 2.4.1, 2.5.1, 2.6.1 and 2.7.1.

### 2.8.1 Inputs Needed\*

#### a. Component Matrix\*

- (1) Rows: One per Component
- (2) Columns: Component ID; Component Type (i.e., Activity Box, Decision Box, Special Event Box, Subprocess Call).

### 2.8.2 Output Produced\*

Same as Inputs Needed.

## GLOSSARY

### Acronyms and Abbreviations

AFLC	Air Force Logistics Command
AFSC	Air Force Systems Command
ATM	Activity Timing and Manpower Data
Box.Proc	The label assigned to the Simulator Event Notice type which processes each Model function box
CCI&C	Code, Compile, Integrate & Checkout
CDR	Critical Design Review
CI	Configuration Item
CPCI	Computer Program Configuration Item
CPDP	Computer Program Development Plan
CPC	Computer Program Component
CPT&E	Computer Program Test & Evaluation
CRISP	Computer Resources Integrated Support Plan
CSDCA	Center for Software Data Collection and Analysis
DELIV	Deliver
DEV	Develop
DID	Data Item Description
DIG	Developmental Integration Group
DOC	Document
DSGN	Designer
ECP	Engineering Change Proposal
ESD	Electronic Systems Division
FACIL	Facility
FCA	Functional Configuration Audit

## GLOSSARY (Continued)

Flow	The label assigned to the Simulator Event Notice type which controls box-to-box transition
FQT	Formal Qualification Testing
FSD	Full-Scale Development
HiSim	High Simulation Level
I&C	Integration and Checkout
LoSim	Low Simulation Level
MGMT	Management
No.Succ	Decision Box Successor List (if "No" Exit is Taken)
ORG	Organize
OTD	Occurrence and Timing Data
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PERT	Program Evaluation and Review Technique
PM	Program Manager
PMR	Program Management Review
PO	Program Office
PQT	Preliminary Qualification Testing
Pred	Function Box Predecessor List
PROC	Procedure
PRGM	Program
PRGMRS	Programmers
PROD	Produce
PROJ	Project

## GLOSSARY (Concluded)

QUAL	Qualification
RFP	Request for Proposal
SARE	Software Acquisition Resource Expenditure
SCEWG	Software Cost Estimation Working Group
SEMP	System Engineering Management Plan
SPEC	Specification
SPRT	Support
SYS	System
TAC	Tactical Air Command
TEMP	Test and Evaluation Master Plan
TIG	Test Integration Group
TOI	Computer Systems Engineering Directorate
Yes.Succ	Function Box Successor List (if "Yes" or Only Exit is Taken)



## REFERENCES

1. AFR 800-2 Acquisition Program Management, AFR 800-2, 14 November 1977.
2. AFR 800-14 Vol. II Acquisition and Support Procedures for Computer Resources in Systems, AFR 800-14, Vol. II, 26 September 1975.
3. CLAPJ76 Judith A. Clapp, A Review of Software Cost Estimation Methods, ESD-TR-76-271, Electronic Systems Division, AFSC, Hanscom AFB, MA, August 1976, ADA029748.
4. DoDD 5000.29 Management of Computer Resources in Major Defense Systems, DoDD 5000.29, 26 April 1976.
5. MIL-STD-483 Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs, MIL-STD-483, 31 December 1970.
6. MIL-STD-1521A Technical Reviews and Audits for Systems, Equipment, and Computer Programs, MIL-STD-1521A, 2 January 1975.
7. SCEWG79 Joseph A. Duquette, AFSC Software Cost Estimation Working Group (SCEWG) Research Management Plan (Draft), Directorate of Cost Analysis, Comptroller, Hq Electronics Systems Division, AFSC, 1 April 1979.